日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: １９９９年　７月２３日

出願番号
Application Number: 平成１１年特許願第２０９８３１号

出願人
Applicant (s): 株式会社東芝

CERTIFIED COPY OF PRIORITY DOCUMENT

２０００年　１月２８日

特許庁長官
Commissioner,
Patent Office 近藤隆彦

出証番号　出証特２０００－３００１１９０

【書類名】　特許願

【整理番号】　A009901940

【特記事項】　特許法第３０条第１項の規定の適用を受けようとする特許出願

【提出日】　平成11年 7月23日

【あて先】　特許庁長官　殿

【国際特許分類】　G06F　7/00

【発明の名称】　演算装置及び暗号処理装置

【請求項の数】　14

【発明者】

【住所又は居所】　東京都府中市東芝町１番地　株式会社東芝府中工場内

【氏名】　斯波　万恵

【発明者】

【住所又は居所】　東京都府中市東芝町１番地　株式会社東芝府中工場内

【氏名】　川村　信一

【特許出願人】

【識別番号】　000003078

【氏名又は名称】　株式会社　東芝

【代理人】

【識別番号】　100058479

【弁理士】

【氏名又は名称】　鈴江　武彦

【電話番号】　03=3502=3181

【選任した代理人】

【識別番号】　100084618

【弁理士】

【氏名又は名称】　村松　貞男

【選任した代理人】

【識別番号】　100068814

【弁理士】

【氏名又は名称】　坪井　淳

【選任した代理人】

【識別番号】　100092196

【弁理士】

【氏名又は名称】　橋本　良郎

【選任した代理人】

【識別番号】　100091351

【弁理士】

【氏名又は名称】　河野　哲

【選任した代理人】

【識別番号】　100088683

【弁理士】

【氏名又は名称】　中村　誠

【選任した代理人】

【識別番号】　100070437

【弁理士】

【氏名又は名称】　河井　将次

【手数料の表示】

【予納台帳番号】　011567

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　演算装置及び暗号処理装置

【特許請求の範囲】

【請求項１】　２の拡大体の多項式基底表現での剰余乗算を実行可能な多倍長の積和演算回路を有する演算装置であって、

前記剰余乗算を乗算処理と剰余算処理とに分割して前記積和演算回路の制御により実行するための制御手段を備えたことを特徴とする演算装置。

【請求項２】　請求項１に記載の演算装置において、

前記積和演算回路は、

前記２の拡大体多項式基底における多項式データを乗算するとき、キャリー伝播をしない単精度の乗算回路と、

前記乗算回路による乗算結果を用いて加算を行う倍精度の加算回路とを有し、

前記制御手段は、前記乗算処理のとき、前記乗算回路と前記加算回路を制御することを特徴とする演算装置。

【請求項３】　請求項２に記載の演算装置において、

前記制御手段に制御され、前記剰余算処理のとき、２つの多項式データの乗算結果を初回の被除多項式データとし、所定の法多項式データを除多項式データとし、前記初回又は２回目以降の被除多項式データと前記除多項式データとに基づいて商計算を行い、上位からバス幅と同じビット数の１ブロックの商多項式データを立てる商立て回路を有し、

前記制御手段は、前記剰余算処理のとき、前記商立て回路の制御により１ブロックの商多項式データが立ったとき、前記乗算回路及び前記加算回路の制御により、前記１ブロックの商多項式データと前記除多項式データとの乗算結果を今回の被除多項式データから減らして次回の被除多項式データを算出し、前記商立て回路の制御から前記被除多項式データの算出までの処理を繰り返して剰余データを得ることを特徴とする演算装置。

【請求項４】　請求項３に記載の演算装置において、

前記商立て回路は、前記商計算のとき、前記除多項式データの上位２ブロックの逆数データと今回の被除多項式データの上位２ブロックとを乗算し、この乗算

結果の上位２ブロック目を前記１ブロックの商多項式データとすることを特徴とする演算装置。

【請求項５】　請求項４に記載の演算装置において、

前記商立て回路は、初回に商多項式データを立てるとき、前記除多項式データの上位２ブロックから逆数データを算出してメモリに記憶させ、２回目以降に商多項式データを立てるとき、前記メモリ内の逆数データを読出して用いることを特徴とする演算装置。

【請求項６】　請求項４又は請求項５に記載の演算装置において、

前記商立て回路は、前記逆数データを算出するとき、前記除多項式データの上位２ブロックのうち、上位から連続した０の数を計数すると、上位から１ブロック＋１ビットの多項式データを最上位ビットを１とするように抽出し、この抽出した多項式データの逆数を求め、得られた逆数の最上位ビット側に、最下位ビットが１で他のビットが０の１ブロックの補正データを連結して全体で２ブロックのデータを求め、このデータを前記計数した０の数だけ上位側にビットシフトさせた結果を前記逆数データとすることを特徴とする演算装置。

【請求項７】　請求項１乃至請求項６のいずれか１項に記載の演算装置を備え、

前記演算装置による２の拡大体の剰余乗算に基づく暗号化又は復号処理を実行することを特徴とする暗号処理装置。

【請求項８】　請求項１乃至請求項６のいずれか１項に記載の演算装置において、

整数型の単位乗算を実行する場合にはキャリーを伝搬させて単位乗算回路を動作させ、２の拡大体の単位乗算を実行する場合にはキャリーを伝搬させずに単位乗算回路を動作させるようにしたことを特徴とする演算装置。

【請求項９】　請求項１乃至請求項６のいずれか１項に記載の演算装置において、

整数型の単位乗算回路と、

前記整数型の単位乗算回路と論理的に隣接して配置された２の拡大体の単位乗算回路と、

前記整数型の単位乗算回路を使用するか、前記２の拡大体の単位乗算回路を使用するかを選択する選択手段と
を備えたことを特徴とする演算装置。

【請求項１０】　請求項１乃至請求項６のいずれか１項に記載の演算装置において、

整数型の単位乗算回路と、

整数型の単位乗算を実行するか、２の拡大体の単位乗算を実行するかの選択信号を前記整数型の単位乗算回路に出力する選択制御手段とを備えると共に、

前記整数型の単位乗算回路は、多倍長の積和演算を実行する際に、整数型の単位乗算を実行すべき旨の選択信号を受けたときにはキャリーを伝播し、２の拡大体の単位乗算を実行すべき旨の選択信号を受けたときにはキャリー伝搬をしないキャリー伝搬制御手段を備え、
前記単位乗算回路におけるキャリー伝搬を制御することにより、整数型乗算と２の拡大体の乗算を切替可能に構成されたことを特徴とする演算装置。

【請求項１１】　請求項１０に記載の演算装置において、

前記キャリー伝搬制御手段は、前記選択信号とキャリーアウト信号を入力とするスイッチによって、１ビット毎の全加算器におけるキャリーの伝搬制御を行うことを特徴とする演算装置。

【請求項１２】　請求項１０に記載の演算装置において、

前記キャリー伝搬制御手段は、１ビット毎の全加算器における２入力ａ，ｂの排他的論理和の結果ｃを加算結果として出力するか、前記結果ｃと入力キャリーとの排他的論理和の結果ｄを加算結果として出力するかを切り替える選択手段からなることを特徴とする演算装置。

【請求項１３】　請求項８乃至請求項１２のいずれか１項に記載の演算装置において、

前記整数型の乗算を実行する場合にはキャリーを伝搬させて加算を実行し、２の拡大体の乗算を実行する場合にはキャリーを伝搬させずに加算を実行する加算回路を備えたことを特徴とする演算装置。

【請求項１４】　請求項８乃至請求項１３のいずれか１項に記載の演算装置

を備え、

前記演算装置による整数型の演算に基づく暗号化又は復号処理と、前記演算装置による２の拡大体の演算に基づく暗号化又は復号処理との双方を切替可能に構成されたことを特徴とする暗号処理装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

この発明は演算装置及び暗号処理装置、特に例えばＩＣカードや情報家電製品に実装される暗号処理用コプロセッサ等に用いるのに適した演算装置及び暗号処理装置に関するものである。

【０００２】

【従来の技術】

公開鍵暗号のＬＳＩ実装においては、従来からＲＳＡ方式等の整数型の演算を行う暗号方式が主に採用されている。この方式では、桁数の大きい整数についての演算を行う必要があるため、ＩＣカード等に応用する場合には専用コプロセッサが必要とされる。このような専用コプロセッサを実装して暗号処理の多倍長整数演算を実現させる方式については既に多数の実績がある。

【０００３】

一方、近年整数型ではなく、２の拡大体（ＧＦ（２＾ｍ）：Ｇａｌｏｉｓ　Ｆｉｅｌｄ（ガロアフィールド））といわれる代数系の上で構成される暗号系，特に２の拡大体の楕円暗号方式が注目されている。

【０００４】

この２の拡大体演算を用いる暗号方式においても、ＲＳＡ方式等の整数型の演算の方式の同様に、扱うビット数を１６０ビット以上と大きくとる必要がある。このため、ＩＣカードのようなＣＰＵの演算力が低い装置上でこれを実現しようとすると比較的処理時間がかかる。したがって、専用のハードウェア（コプロセッサ）を用いて高速化したいという要求がある。

【０００５】

このようにＲＳＡ方式であっても、楕円暗号方式であっても、ＩＣカード等に

おいて高速な暗号処理を実現させるためには専用の演算用コプロセッサを用意する必要がある。

【０００６】

図２３は暗号処理用のコプロセッサを含むＩＣカード用ＬＳＩのブロック図である。

【０００７】

同図に示す構成要素を含む専用ＬＳＩが例えばＩＣカードに組み込まれる。このＬＳＩにおいては、ＣＰＵ、ＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭが１チップに収容され、コプロセッサはＲＡＭ、演算部、制御部から構成される。コプロセッサはＣＰＵの制御下にあって多倍長のべき乗剰余演算や四則演算といった公開鍵暗号の基本演算に関しＣＰＵを補助する。つまり、暗号の基本演算を行う部分は図１０におけるコプロセッサであり、この部分がどのように構成されているかが問題となる。

【０００８】

図２４は図２３に示すＬＳＩにおけるコプロセッサ部分の構成例を示すブロック図である。

【０００９】

ＲＳＡ方式では整数型の演算が行われるため、このコプロセッサは図２４に示すように整数型の乗算器として構成されている。

【００１０】

一方、楕円暗号方式のＬＳＩを作成する場合、その全体的な構成は図２３に示すＬＳＩと同様あるいはこれと類似する構成のものとなるが、コプロセッサ部分における演算は整数型の演算でなく、２の拡大体演算を行うコプロセッサを用意する必要がある。

【００１１】

図２５は多項式基底の２の拡大体演算を行うコプロセッサのハードウェア構成例を示すブロック図である。

【００１２】

同図は、「楕円暗号のハードウェア実装，ＳＣＩＳ’９８－１０．１．Ｃ」に

て発表された特殊な既約多項式を用いる円分体といわれる２の拡大体の一種の演算装置を示している。この演算装置は、２の拡大体上の加算、二乗、乗算及び逆元演算を実行できる構成を備えており、これによって楕円曲線上の点の演算に必要な２の拡大体の演算を実行する。このような演算装置をＩＣ化すれば、図２３のＬＳＩに適用し得る２の拡大体演算用のコプロセッサになる。

【００１３】

ここで、加算回路および二乗算回路はｍ個のＥＸ－ＯＲで構成され、乗算回路８１は図２６に示す回路構成で実現する。

【００１４】

図２６は円分体といわれる２の拡大体の乗算回路を示す図である。

【００１５】

この乗算回路８１は、ｍビットの入力レジスタＡ，Ｂを持ち多項式ａ（ｘ）の係数を入力レジスタＡに固定値として入力し、入力レジスタＢからは多項式ｂ（ｘ）の係数を最上位ビットから１クロック毎にシフトしながら演算していく。同図におけるブロックＤはフィードバックレジスタを構成するフリップフロップである。ｍ回シフトした時点での各ブロックＤの値が出力レジスタＣに読み出され、ａ（ｘ）＊ｂ（ｘ）の演算結果となる。

【００１６】

図２４と図２６夫々の回路を見比べてもわかるように、乗算と一口に言っても整数型乗算と多項式基底の２の拡大体乗算とでは、その乗算を実行するためのアーキテクチャーが全く異なる。したがって、従来は暗号方式毎にハードウェアを構成しようとする試みがなされてきた。

【００１７】

一方、楕円曲線暗号の基本演算における２の拡大体の剰余乗算には、拡大体ＧＦ（q^m）上の多項式ｆ（ｘ）による割り算回路としての線形フィードバックシフトレジスタ（ＬＦＳＲ）を用いた演算装置が広く用いられている。なお、法多項式ｆ（ｘ）は、次式の通りである。

【００１８】

$$f(x)=f_m x^m+f_{m-1} x^{m-1}+\cdots+f_1 x+f_0,\quad f_m=1$$

図２７は係る線形フィードバックシフトレジスタＬＦＳＲの構成を示すブロック図である。このＬＦＳＲ９０は、入力側からEX-OR の加算器９１$_1$～９１$_m$と１クロックの遅延素子（以下、レジスタという）９２$_1$～９２$_m$とが交互に縦続接続されており、m個目のレジスタ９２$_m$から得られる出力が抽出されて夫々係数器９３$_1$～９３$_m$を介して個別にm個の加算器９１$_1$～９１$_m$にフィードバックされる構成となっている。

【００１９】

このＬＦＳＲ９０は、単位時間（クロック）毎に動作する。なお、シフトレジスタにおいて、動作用のクロックパルスを１クロック進めることをシフトするといい、シフトレジスタに組込まれるレジスタ９２$_1$～９２$_m$の数mをシフトレジスタの段数という。

【００２０】

ここで、q＝２のとき、各レジスタ９２$_1$～９２$_m$は、１ビットのフリップフロップが適用可能である。各係数器９３$_1$～９３$_m$は、“１”又は“０”を乗じるものであり、１を乗じる場合には結線されるが、０を乗じる場合には結線されない。また、各加算器９１$_1$～９１$_m$は、２入力のＥＸ－ＯＲが使用されている。

【００２１】

このようなＬＦＳＲ９０では、入力側（左側）から被除多項式の係数が高次の項から順に入力されると、出力側（右側）から商多項式の係数が高次の項から順に出力される。ここで、被除多項式の０次の項を入力し終えたときの各レジスタ（フリップフロップ）９２$_1$～９２$_m$の内容が剰余多項式の係数となる。

【００２２】

しかしながら、以上のＬＦＳＲ９０を用いた演算装置では、拡大次数mのビット数と同数のレジスタ９２$_1$～９２$_m$を必要とし、レジスタ９２$_1$～９２$_m$の構成が拡大次数mにより制約を受けている。このため、拡大次数mが増加すると、演算装置ごと作り直す必要が生じてしまう。

【００２３】

【発明が解決しようとする課題】

上記したように、楕円暗号方式は現在注目されてはいるものの、現状ではＲＳＡ暗号方式が未だ主流であるため、楕円暗号方式を用いるＩＣカードにおいてもＲＳＡ暗号にも対応させたいという要請が強い。

【００２４】

ここで、従来の整数型の暗号と２の拡大体の暗号を同一ＩＣカードに実装しようとした場合、上記した従来技術の延長ではそれぞれに対応するコプロセッサを搭載する必要が生じる。しかしながら、２つのコプロセッサを搭載したのでは、面積制約の大きいＩＣカードにおいてそのチップ面積を圧迫するという問題が生じる。

【００２５】

一方、２の拡大体の剰余乗算においては、拡大次数ｍの増加により、演算装置ごと作り直す必要が生じるというハードウェア的な制限がある。

【００２６】

本発明はこのような実情を考慮してなされたもので、２の拡大体の拡大次数ｍを増加しても、装置本体を作り直さずに演算を実行し得る演算装置及び暗号処理装置を提供することを目的とする。

【００２７】

また、本発明の他の目的は、最小のアーキテクチャを追加するだけで整数型の演算に加えて２の拡大体上の演算をも実行できる演算装置及び暗号処理装置を提供することにある。

【００２８】

【課題を解決するための手段】

上記課題を解決するために、請求項１に対応する発明は、２の拡大体の多項式基底表現での剰余乗算を実行可能な多倍長の積和演算回路を有する演算装置であって、剰余乗算を乗算処理と剰余算処理とに分割して積和演算回路の制御により実行するための制御手段を備えた演算装置である。

【００２９】

本発明はこのような手段を設けたので、剰余算処理の際に線形フィードバックシフトレジスタではなく、多倍長の積和演算回路が演算を行うので、１以上の任

意の拡大次数を使用でき、２の拡大体の拡大次数を増加しても、装置本体を作り直さずに演算を実行することができる。

【００３０】

次に、請求項２に対応する発明は、請求項１に対応する演算装置において、積和演算回路としては、２の拡大体多項式基底における多項式データを乗算するとき、キャリー伝播をしない単精度の乗算回路と、乗算回路による乗算結果を用いて加算を行う倍精度の加算回路とを有し、制御手段としては、乗算処理のとき、乗算回路と加算回路を制御する演算装置である。

【００３１】

本発明はこのような手段を設けたので、請求項１と同様の作用を容易且つ確実に奏することができる。

【００３２】

次に、請求項３に対応する発明は、請求項２に対応する演算装置において、制御手段に制御され、剰余算処理のとき、２つの多項式データの乗算結果を初回の被除多項式データとし、所定の法多項式データを除多項式データとし、初回又は２回目以降の被除多項式データと除多項式データとに基づいて商計算を行い、上位からバス幅と同じビット数の１ブロックの商多項式データを立てる商立て回路を有し、制御手段としては、剰余算処理のとき、商立て回路の制御により１ブロックの商多項式データが立ったとき、乗算回路及び加算回路の制御により、１ブロックの商多項式データと除多項式データとの乗算結果を今回の被除多項式データから減らして次回の被除多項式データを算出し、商立て回路の制御から被除多項式データの算出までの処理を繰り返して剰余データを得る演算装置である。

【００３３】

この演算装置では、１ブロックの商多項式データと除多項式データとの乗算結果が毎回（ｍ＋１）ブロックになる。

また、この乗算結果を今回の被除多項式から減算（＝加算）して、（２ｍ－１＊ｎ）ブロックの次回の被除多項式データを算出し（ｎは乗算回数）、すなわち前回の被除多項式データを１ブロックずつ減らしていく。

【００３４】

本発明はこのような手段を設けたので、請求項１又は請求項２と同様の作用に加え、ハードウェアの特性を生かして剰余算と商計算の効率化を図ることができる。

【００３５】

次に、請求項４に対応する発明は、請求項３に対応する演算装置において、商立て回路としては、商計算のとき、除多項式データの上位２ブロックの逆数データと今回の被除多項式データの上位２ブロックとを乗算し、この乗算結果の上位２ブロック目を１ブロックの商多項式データとする演算装置である。

【００３６】

本発明はこのような手段を設けたので、請求項３と同様の作用に加え、得られた商多項式のうち、有効数字の部分を抽出できるので、演算精度の最適化を図ることができる。

【００３７】

次に、請求項５に対応する発明は、請求項４に対応する演算装置において、商立て回路としては、初回に商多項式データを立てるとき、除多項式データの上位２ブロックから逆数データを算出してメモリに記憶させ、２回目以降に商多項式データを立てるとき、メモリ内の逆数データを読出して用いる演算装置である。

【００３８】

本発明はこのような手段を設けたので、同じ法多項式下での重複する剰余算の実行時に、逆数データをメモリから読出して商を立てるので、２回目の商計算以降は逆数データの算出時間を省略でき、もって、２の拡大体演算の処理時間を短縮することができる。また、逆数データを事前に計算できるので、２の拡大体の剰余乗算を、乗算と加算とを行う積和演算回路のみを用いて実現することができる。

【００３９】

次に、請求項６に対応する発明は、請求項４又は請求項５に対応する演算装置において、商立て回路としては、逆数データを算出するとき、除多項式データの上位２ブロックのうち、上位から連続した０の数を計数すると、上位から１ブロック＋１ビットの多項式データを最上位ビットを１とするように抽出し、この抽

出した多項式データの逆数を求め、得られた逆数の最上位ビット側に、最下位ビットが１で他のビットが０の１ブロックの補正データを連結して全体で２ブロックのデータを求め、このデータを前記計数した０の数だけ上位側にビットシフトさせた結果を逆数データとする演算装置である。

【００４０】

本発明はこのような手段を設けたので、一般的な多倍長の整数型の除算で用いられる単精度除算を用いたKnuth のアルゴリズム（文献：Knuth, D. E. The Art of Computer Programming, Vol.2, Reading, Mass.: Addison Wesley, 2^{nd} edition, (1981)）による除数の正規化や近似商の補正処理、商や剰余といった演算結果の逆正規化の処理を行わない観点から、予め補正した値を逆数データとしたので、ビットシフトの回数を減らすことができ、演算装置を最適化することができる。

【００４１】

次に、請求項７に対応する発明は、請求項１乃至請求項６のいずれか１項に対応する演算装置を備え、演算装置による２の拡大体の剰余乗算に基づく暗号化又は復号処理を実行する暗号処理装置である。

【００４２】

本発明はこのような手段を設けたので、楕円曲線暗号等の２の拡大体の剰余乗算に基づく暗号化又は復号処理を実行することができる。

【００４３】

次に、請求項８に対応する発明は、請求項１乃至請求項６のいずれか１項に対応する演算装置であって、整数型の単位乗算を実行する場合にはキャリーを伝搬させて単位乗算回路を動作させ、２の拡大体の単位乗算を実行する場合にはキャリーを伝搬させずに単位乗算回路を動作させるようにした演算装置である。

【００４４】

本発明はこのような手段を設けたので、請求項１乃至請求項６のいずれかの作用に加え、最小のアーキテクチャを追加するだけで整数型の演算に加えて２の拡大体上の演算をも実行することができる。

【００４５】

次に、請求項９に対応する発明は、請求項１乃至請求項６のいずれか１項に対応する演算装置において、整数型の単位乗算回路と、整数型の単位乗算回路と論理的に隣接して配置された２の拡大体の単位乗算回路と、整数型の単位乗算回路を使用するか、２の拡大体の単位乗算回路を使用するかを選択する選択手段とを備えた演算装置である。

【００４６】

本発明はこのような手段を設けたので、請求項１乃至請求項６のいずれかの作用に加え、２の拡大体の単位乗算回路を追加するだけで整数型の乗算と２の拡大体の乗算の双方を実行することができる。

【００４７】

次に、請求項１０に対応する発明は、請求項１乃至請求項６のいずれか１項に対応する演算装置において、整数型の単位乗算回路と、整数型の単位乗算を実行するか、２の拡大体の単位乗算を実行するかの選択信号を整数型の単位乗算回路に出力する選択制御手段を備えるとともに、整数型の単位乗算回路は、多倍長の積和演算を実行する際に、整数型の単位乗算を実行すべき旨の選択信号を受けたときにはキャリーを伝播し、２の拡大体の単位乗算を実行すべき旨の選択信号を受けたときにはキャリー伝搬をしないキャリー伝搬制御手段を備え、単位乗算回路におけるキャリー伝搬を制御することにより、整数型乗算と２の拡大体の乗算を切替可能に構成された演算装置である。

【００４８】

本発明はこのような手段を設けたので、請求項１乃至請求項６のいずれかの作用に加え、キャリー伝搬制御手段を追加するだけで整数型の乗算と２の拡大体の乗算の双方を実行することができる。

【００４９】

次に、請求項１１に対応する発明は、請求項１０の演算装置において、キャリー伝搬制御手段は、選択信号とキャリーアウト信号を入力とするスイッチによって、１ビット毎の全加算器におけるキャリーの伝搬制御を行う演算装置である。

【００５０】

本発明はこのような手段を設けたので、選択信号とキャリーアウト信号を入力

とするスイッチにより請求項１０に係る発明を実現させることができる。

【００５１】

次に、請求項１２に対応する発明は、請求項１０の演算装置において、キャリー伝搬制御手段は、１ビット毎の全加算器における２入力ａ，ｂの排他的論理和の結果ｃを加算結果として出力するか、結果ｃと入力キャリーとの排他的論理和の結果ｄを加算結果として出力するかを切り替える選択手段からなる演算装置である。

【００５２】

本発明はこのような手段を設けたので、選択手段により請求項１０に係る発明を実現させることができる。

【００５３】

次に、請求項１３に対応する発明は、請求項８～１２の演算装置において、整数型の乗算を実行する場合にはキャリーを伝搬させて加算を実行し、２の拡大体の乗算を実行する場合にはキャリーを伝搬させずに加算を実行する加算回路を備えた演算装置である。

【００５４】

本発明はこのような手段を設けたので、請求項８～１２のいずれかの作用に加え、積和演算における加算部分についても整数型の乗算と２の拡大体の乗算との双方を確実に実行することができる。

【００５５】

次に、請求項１４に対応する発明は、請求項８～１３の何れかの演算装置を備え、演算装置による整数型の演算に基づく暗号化又は復号処理と、演算装置による２の拡大体の演算に基づく暗号化又は復号処理との双方を切替可能に構成された暗号処理装置である。

【００５６】

本発明はこのような手段を設けたので、請求項８～１３のいずれかの作用に加え、ＲＳＡ暗号等の整数型の演算に基づく暗号と、楕円曲線暗号等の２の拡大体の演算に基づく暗号の双方の処理を行うことができる。

【００５７】

【発明の実施の形態】

以下、本発明の各実施形態について図面を用いて説明する。

（第１の実施形態）

図１は本発明の第１の実施形態に係る演算装置の構成例を示すブロック図である。

コプロセッサ１として構成される本実施形態の演算装置は、整数型乗算及び２の拡大体乗算の双方の演算が可能な多倍長積和乗算装置であり、この乗算処理を仕方を制御することにより、加算、二乗あるいは逆元等の他の演算を実行するものである。また、本演算装置がＬＳＩ等に組み込まれることによってＲＳＡ暗号及び楕円暗号の双方が実現可能な暗号処理装置が構成される。ここで組込対象となるＬＳＩは例えば図２３に示すような装置である。

【００５８】

このコプロセッサ１において演算部４は制御部５によってコントロールされ、演算途中のデータを格納するメモリ２から接続される３２ビットのデータバス３からデータを入出力するようになっている。

【００５９】

データバス３からの入力データはバッファＺ，Ｙ，Ｘに格納され、データバス３への出力データはバッファＲに格納されるようになっている。

【００６０】

入力データＸ及びＹは乗算対象となるデータであり、このうちデータＹは一度に多数桁の乗算となるのを回避するために所定桁毎に分割されたデータとして入力される。一方、データＺは乗算を複数回に分けて実行するために生じる途中結果であり、これをＸＹの乗算結果に足し、さらにその和の結果にキャリーＣと言われる桁上がり部分を足して１サイクルの乗算が終了する。その結果からキャリーを除いたデータＲがバッファＲを介してデータバス３に出力され、次のサイクルの演算にデータＺとして使用される。このサイクルを複数回繰り返すことにより多倍長整数乗算あるいは２の拡大体乗算（厳密には後述のｃ’）の乗算が実現される。

【００６１】

また、コプロセッサ１は、上記演算を実現するために、バッファＸ，Ｙ，Ｚ，Ｒの他、整数型乗算回路１１，２の拡大体上乗算回路１２，セレクタ１３，加算回路１４，加算回路１５，キャリー保持部１６及び制御部５を備えている。

【００６２】

整数型乗算回路１１は、バッファＸ内のデータＸとバッファＹ内のデータＹとを整数型乗算し、その結果をセレクタ１３に出力する。

【００６３】

２の拡大体上乗算回路１２は、バッファＸ内のデータＸとバッファＹ内のデータＹとにより２の拡大体上乗算の一部（ｃ’）を実行し、その結果をセレクタ１３に出力する。

【００６４】

セレクタ１３は、制御部５からの信号Ｓ１に従って、整数型乗算回路１１又は２の拡大体上乗算回路１２からの出力の何れかを加算回路１４に出力する。

【００６５】

加算回路１４は全加算器からなり、バッファＺ内のデータＺとセレクタ出力を加算して加算回路１５に出力する。この加算回路１４においては、整数型の加算と２の拡大体の加算との切替が制御信号Ｓ１に従って行われるようになっている。なお、この加算切替については後述する。

【００６６】

加算回路１５は、加算回路１４の出力にキャリー保持部１６に保持されたキャリーＣを加算し上位３２ビットを次のキャリーＣとしてキャリー保持部１６に出力し、下位８ビットをこのサイクルの演算結果であるデータＲとしてバッファＲに出力する。なお、加算回路１５においても、制御信号Ｓ１により、整数型の加算と２の拡大体の加算との切替が行われるようになっている。

【００６７】

キャリー保持部１６は、加算回路１５から出力されたキャリーＣを保持し、次の演算サイクルにおいて保持したキャリーＣを加算回路１５に与える。

【００６８】

制御部５は、整数演算制御部２１と２の拡大体演算制御部２２からなり、これ

らの何れかのコマンド群に従って演算部を制御する。このコマンド切り替えは、外部のＣＰＵ（例えば図２３に示すＣＰＵ）からの指示によって行われる。

【００６９】

整数演算制御部２１は、演算部４を多倍長整数演算型の乗算器として動作するように制御するものである。このために、制御信号Ｓ１によりセレクタ１３が整数型単精度乗算器１１の出力を加算回路１４に出力するように制御するとともに、加算回路１４及び１５を整数型加算回路として動作するよう制御する。さらに、整数型乗算器として演算部４の動作を制御することで他の四則演算などの演算処理の実行する。

【００７０】

また、２の拡大体演算制御部２２は、演算部４を２の拡大体乗算器として動作するように制御するものである。このために、制御信号Ｓ１によりセレクタ１３が２の拡大体型単精度乗算器１１の出力を加算回路１４に出力するように制御するとともに、加算回路１４及び１５を２の拡大体型加算回路として動作するように制御する。さらに、２の拡大体型乗算器として演算部４の動作を制御することで加算、二乗算を実現する。

【００７１】

なお、制御部５からは、上記した各処理を実現するため、制御信号Ｓ２を出力して各部を制御する。

【００７２】

次に、以上のように構成された本実施形態における演算装置の動作について説明する。

【００７３】

この演算装置（コプロセッサ１）は、整数型の乗算装置に乗算回路１２，セレクタ１３等を組み込むことにより、２の拡大体の乗算装置としての処理を実現可能とするものである。ここで、２の拡大体では、以下に示すようにｍ－１次の多項式をｍビットのベクトル表現で表すことができる。

【００７４】

$$a(x) = a_{m-1} x^{m-1} + a_{m-2} x^{m-2} + \cdots + a_1 x + a_0 \quad \cdots (1)$$

$$= [a_{m-1}, \cdots, a_1, a_0]$$

$$b(x) = b_{m-1}x^{m-1} + b_{m-2}x^{m-2} + \cdots + b_1 x + b_0 \quad \cdots (2)$$

$$= [b_{m-1}, \cdots, b_1, b_0]$$

ここで、２の拡大体の乗算はＧＦ（2^m）上のｍ次の規約多項式ｆ（ｘ）をモジュラスとする剰余乗算である。また、２の拡大体の二つの元ａ（ｘ）とｂ（ｘ）の積ｃ（ｘ）は、次のように定義されている。

【００７５】

$$c(x) = a(x) \cdot b(x) \bmod f(x) \quad \cdots (3)$$

$$= \Sigma a_k \cdot x^k \cdot b(x) \bmod f(x)$$

$$= c_{m-1}x^{m-1} + c_{m-2}x^{m-2} + \cdots + c_1 x + c_0$$

$$= [c_{m-1}, \cdots, c_1, c_0]$$

また、法多項式ｆ(x) は、次式で表せる。

$$f(x) = f_m x^m + f_{m-1}x^{m-1} + \cdots + f_1 x + f_0 \quad \cdots (4)$$

$$= [f_m, f_{m-1}, \cdots, f_1, f_0]$$

２の拡大体の多項式の乗算は、図２６に示すように乗数のサイクルシフトによるシフトレジスタを構成し、ｍサイクルシフト後の剰余多項式を乗算結果とするのが一般的であるが、本実施形態では整数型の暗号処理ＬＳＩで広く使われている多倍長の積和演算回路に若干の変更を加えて処理する。

【００７６】

なお、制御部５からの制御信号Ｓ１により、コプロセッサ１が整数型の演算装置として動作するときには、同演算装置は多倍長積和演算回路として機能している。この多倍長積和演算回路において、制御信号Ｓ１による切替により、２の拡大体上乗算回路１２において２の拡大体の乗算の一部分である（５）式が計算される。

【００７７】

$$c'(x) = a(x) \cdot b(x) \quad \cdots (5)$$

なお、２の拡大体上乗算回路１２ではｃ’を計算する段階においては（６）式における「ｃ（ｘ）’ ｍｏｄ ｆ（ｘ）」の部分は計算されない。すなわちｃ’自体は、制御信号Ｓ１により乗算回路１２及び加算回路１４，１５を切り替える

のみで、整数型乗算における２つの数の積と全く同様に演算される。

【００７８】

なお、ｃ’（ｘ）＝ａ（ｘ）・ｂ（ｘ）においてｍビットの乗数、被乗数は３２ビットに分割されてメモリから読み出され、演算結果は３２ビット毎にメモリに書き込まれる。この時、最終的な演算結果は２ｍビットとなる。

【００７９】

整数型乗算回路１１による整数演算と２の拡大体上乗算回路１２による２の拡大体多項式演算の違いは、桁上がりの有無である。整数演算では足し算の論理式は

０＋０＋Ｃａｒｒｙ（＝０）＝０、Ｃａｒｒｙ＝０

１＋０＋Ｃａｒｒｙ（＝０）＝１、Ｃａｒｒｙ＝０

１＋１＋Ｃａｒｒｙ（＝０）＝０、Ｃａｒｒｙ＝１

という様に下位ビットのキャリーを考慮した演算をしなければならないのに対し、２の拡大体の代数系においては、各ビットが多項式における次数の係数を示しているため異なる次数への桁上がりを考慮しなくてもよい。

【００８０】

このことに着目して本実施形態では整数型演算器（乗算器や加算器）において、キャリー伝播を許す通常のモードと、キャリー伝播を実行しないモードを切り替えて使えるようにしているのである。ここでキャリー伝播を許さない（実行しない）モードは２の拡大体演算を行うのに用いられる。なお、キャリー伝播のモードを切り替えのために追加すべき回路は全体の回路規模に比べわずかである。

【００８１】

図２はｃ’（ｘ）＝ａ（ｘ）・ｂ（ｘ）を実現するための４＊４ビットの単位乗算の回路構成例を示す図である。

【００８２】

同図の単位演算装置を８＊３２ビット構成にしたものが図１における２の拡大体上乗算回路１２である。なお、図２（ｂ）の回路は同図（ａ）の回路の入力部分２９を示すものである。

【００８３】

一方、図３は整数型乗算を実現するための４＊４ビットの単位乗算の回路構成例を示す図である。

【００８４】

同図の単位演算装置を８＊３２ビット構成にしたものが図１における整数型乗算回路１１である。なお、図３（ａ）に用いられる全加算器ＦＡの構成は図３（ｃ）に示され、さらに図３（ｃ）に示す全加算器ＦＡのキャリー３１の構成が同図（ｄ）に示されている。また、図３（ｂ）の回路は同図（ａ）の回路の入力部分３０を示すものである。

【００８５】

本実施形態の演算装置では、２の拡大体上乗算回路１２と整数型乗算回路１１とが論理的に隣接して配置されており、制御部５の２の拡大体演算コマンドから生成される制御信号Ｓ１により整数型、２の拡大体型のいずれかの乗算回路１１，１２が選択されて処理が行われる。

【００８６】

セレクタ１３の出力は次段の加算回路１４に入力される。ここでＺ＋（Ｙ＊Ｘ）加算回路１４は４０ビットのデータ（Ｙ＊Ｘ）と８ビットのデータＺの全加算器だが、ここでも前述の制御信号により各ビットの加算結果のキャリーを次段へ伝播しないスイッチを付加することにより２の拡大体の加算が実現される。

【００８７】

図４は本実施形態におけるコプロセッサに用いられるキャリー制御機能付きの４ビットのリップルキャリー型全加算器の構成例を示すブロック図である。

【００８８】

このような構成の全加算器を、４０ビットデータと８ビットデータとの加算が可能となるように拡張したものが図１の加算回路１４である。

【００８９】

また、図４の回路において、各全加算器３２の間にはスイッチ３３が設けられ、キャリーの伝搬を制御できるようになっている。

【００９０】

図５は本実施形態の加算回路に用いられる全加算器及びキャリー制御スイッチ

の構成例を示す図である。

【００９１】

この全加算器３２及びスイッチ３３は、１ビット分のキャリー制御機能付き全加算器４２を構成している。ここで、全加算器３２は、図３（ｃ）に示す全加算器ＦＡと同様に構成され、全加算器３２内のキャリー３１は図３（ｄ）に示すキャリーと同様に構成されている。

【００９２】

また、全加算器３２間のキャリー伝搬ラインに設けられたスイッチ３３は制御部５からの制御信号Ｓ１によって制御され、整数型演算を行うときには接続され、２の拡大体演算を行うときには遮断される。

【００９３】

以上のように構成された加算回路１４からの出力（Ｚ＋（Ｙ＊Ｘ））は加算回路１５に引き渡される。

【００９４】

すなわち、演算ブロック最終段のＣ＋Ｚ＋（Ｙ＊Ｘ）加算回路１５によって、乗算結果の４０ビットの下位８ビットがデータＲとして出力され、上位３２ビットが次のサイクルのＺ＋（Ｙ＊Ｘ）に足し込まれる。

【００９５】

ここで、加算回路１５は加算回路１４と同様に、前述の制御信号Ｓ１により制御される図４に示すキャリー制御機能付き全加算器であるので、整数型ではＬＳＢに桁あわせをした全加算器として整数型加算が実行され、２の拡大体演算では２の拡大体加算が実行される。

【００９６】

加算回路１５の出力データＲはデータバス３を介して一旦外部のメモリ２に出力され、再びデータＺとなってコプロセッサ１内に戻り整数型乗算若しくは２の拡大体上の乗算が継続され、必要なサイクル数だけ繰り返されて乗算結果が得られる。

【００９７】

ここで２の拡大体の乗算コマンドでは、（５）式の結果が得られるが、２の拡

大体乗算は（６）式に示す定義通り、既約多項式ｆ（ｘ）をモジュラスとする剰余演算によって完結する。剰余演算は割り算の筆算同様、被除数の上位桁から商を立て現在の商と除数をかけたものから現在の被除数を引く（２の拡大体では減算は加算と同じ）処理を必要なサイクル数だけ繰り返せばよく、２の拡大体の乗算コマンドと加算コマンドを実行することによって実現できる（詳細は第３の実施形態で述べる）。２の拡大体の二乗算は乗算と同じ処理で実現でき、逆元計算は、乗算と二乗算を相互に繰り返すことにより実現できる。

【００９８】

一例として、２の拡大体の加算コマンドに従って演算部４が２の拡大体の加算装置として機能する場合を説明する。

【００９９】

２の拡大体上の加算は、通常の多項式の加算と同じで、同じ次数の係数同士の足し算を行う。

【０１００】

$$c(x) = a(x) + b(x) \qquad \cdots (7)$$

$$= [a_{m-1} + b_{m-1},\ a_{m-2} + b_{m-2},\ \cdots,\ a_0 + b_0]$$

このとき、各次数の係数の和は０＋０＝１＋１＝０、０＋１＝１＋０＝１となり、整数型加算のようにキャリーは発生しない。従って、２の拡大体での加算は、一般にはｍ個のＥＸ－ＯＲで実装できることになる。

【０１０１】

整数型の乗算装置において加算はｃ＝ｂ＋ａ＊１として扱えるので、本実施形態における２の拡大体の加算もこのアルゴリズムをそのまま利用し、ｃ（ｘ）＝ｂ（ｘ）＋ａ（ｘ）＊１として実行する。この演算は加算回路１４，１５に図４の全加算器が用いられていることから、制御信号Ｓ１の切替で実現できる。

【０１０２】

また、制御信号Ｓ１による切替でコプロセッサ１は図２４に示すコプロセッサと同様な機能を持つ回路となり、整数型演算も実現される。

【０１０３】

上述したように、本発明の実施の形態に係る演算装置は、整数型乗算装置に、

整数型乗算の単位乗算装置と回路構成の似ている２の拡大体乗算の単位演算装置とを設け、整数型の乗算コマンドに２の拡大体演算コマンドとを追加し、２の拡大体演算コマンドから生成される制御信号により制御されるセレクタと、全加算器の各ビットのキャリーの伝播を制御するスイッチの追加するようにしたので、従来型のシフトレジスタによるシーケンシャルな２の拡大体の乗算装置を用いることなく整数および２の拡大体演算の両方を実行することができる。

【０１０４】

したがって、従来からある整数型の演算器への追加拡張機能として、ごく少ない命令と回路の追加することにより、多倍長の積和演算回路で２の拡大体の加算、乗算を実行することが可能な公開鍵暗号処理用アクセラレータを提供することができる。なお、本実施形態を実現するのに、必要な回路追加の量は全体の回路規模に比べてわずかである。

【０１０５】

本実施形態の暗号処理装置によれば、暗号処理用コプロセッサとして、整数型のＲＳＡ方式に加え２の拡大体の楕円暗号方式も処理できる豊富な機能をもつＬＳＩを特に実装面積を増大させることなく提供できる。したがって、ＩＣカードのような実装可能面積の少ない装置において、ＲＳＡ、楕円暗号の双方を処理できる暗復号装置を実現させることができる。

【０１０６】

（変形例１）

本変形例では、図４に示す加算回路１４，１５を構成するキャリー制御機能付き全加算器について説明する。

【０１０７】

図６はキャリー制御機能付き全加算器の変形例を示す図である。

【０１０８】

このキャリー制御機能付き全加算器４３は、スイッチ３３と全加算器３２から構成される点で図５の回路と共通する。しかし、図５の回路ではキャリー３１の出力側にスイッチ３３が設けられているのに対し、図６の回路ではキャリー３１の入力側にスイッチ３３が設けられている。

【０１０９】

（変形例２）

本変形例では、更に他のキャリー制御機能付き全加算器について説明する。

【０１１０】

図７はキャリー制御機能付き全加算器の他の変形例を示す図である。

【０１１１】

このキャリー制御機能付き全加算器４４は、加算結果の出力選択を制御することによりキャリー制御を行う。すなわちスイッチ３３’はセレクタで構成され、このセレクタは制御信号Ｓ１に基づき、ＥＸＯＲ３５又はＥＸＯＲ３６の出力を選択する。これを複数個連結したリップルキャリー型加算器は、制御信号Ｓ１によりキャリー伝播の有無を制御できる。

【０１１２】

図７の制御信号Ｓ１を２の拡大体演算コマンドによる制御信号とするとＳ１が“１”のときａとｂのＥＸＯＲ３５の出力が演算結果となり、２の拡大体の加算装置として機能し、Ｓ１が“０”のとき全加算器の出力が演算結果となって整数型の加算装置として機能する。

【０１１３】

（第２の実施形態）

図８は本発明の第２の実施形態に係る演算装置の構成例を示すブロック図であり、図１と同一部分には同一符号を付して説明を省略し、ここでは異なる部分についてのみ述べる。なお、以下の各実施形態も同様にして重複した説明を省略する。

【０１１４】

この演算装置であるコプロセッサ１’は、図１における整数型乗算回路１１，２の拡大体上乗算回路１２及びセレクタ１３に代えて乗算回路４１を備える他、第１の実施形態と同様に構成されている。

【０１１５】

この乗算回路４１は、制御部５からの制御信号Ｓ１によって整数型乗算と２の拡大体上乗算（（６）式のｃ’のみ）を切り替えるようになっている。

【０１１６】

図９は本実施形態の乗算回路を実現するための４＊４ビットの単位乗算の回路構成例を示す図である。なお、現実の乗算回路４１は、同図の単位演算装置を８＊３２ビット構成にしたものである。また、図９（ｂ）の回路は同図（ａ）の回路の入力部分２９を示すものである。

【０１１７】

この乗算回路４１は、図９（ａ）に示すように、全加算器として図５に示すキャリー制御機能付き全加算器４２を用いているので、制御信号Ｓ１に従ってキャリー伝搬の有無を制御できる。したがって、２の拡大体演算コマンドによる整数型乗算と２の拡大体上乗算との切替が実現される。

【０１１８】

こうして本実施形態の演算装置では第１の実施形態と同様な動作が実現される。

【０１１９】

上述したように、本発明の実施の形態に係る演算装置及び暗号処理装置は、整数型乗算回路１１，２の拡大体上乗算回路１２及びセレクタ１３に代えて乗算回路４１を用いるようにし、一つの回路４１で回路１１，１２及び１３の機能を実現するようにしたので、第１の実施形態と同様な効果が得られる他、より少ない回路追加で整数型乗算と２の拡大体上乗算との切り替えを実現にすることができる。

【０１２０】

なお、本実施形態ではキャリー制御機能付き全加算器４２として図５に示すものを用いるようにしたが、キャリー制御機能付き全加算器４２に代えて、図６又は図７に示すキャリー制御機能付き全加算器４３又は４４を用いるようにしてもよい。

【０１２１】

（第３の実施形態）

図１０は本発明の第３の実施形態に係る演算装置及び暗号処理装置に適用されるコプロセッサの構成例を示すブロック図である。

【０１２２】

本実施形態は、第１の実施形態に関し、剰余算の部分を具体的に示す具体例であり、図示するように、制御部５において、前述した機能に剰余算機能が付加された２の拡大体演算制御部２２ａと、この剰余算機能に制御され、且つ逆数計算部５１を有する商立て回路５０とを備えている。

【０１２３】

ここで、２の拡大体演算制御部２２ａは、（５）式の乗算結果ｃ’（ｘ）を得るために演算部４を制御する前述した機能に加え、この乗算結果ｃ’（ｘ）に対して法多項式ｆ(x) による剰余算を実行させるように、演算部４及び商立て回路５０を制御する機能をもっている。具体的には、制御機能は、後述する演算アルゴリズムに基づいて、メモリ２やバッファＸ，Ｙ，Ｚ，Ｒとの間でデータを入出力する機能と、この入出力に連動し、乗算コマンド、加算コマンド及び逆数計算コマンドなどの各種コマンドを生成して対応する演算回路に与える機能とを有している。

【０１２４】

商立て回路５０は、数式的には剰余算のうちで、被除多項式ｃ’（ｘ）を法多項式ｆ（ｘ）で除した商を算出するためのものであり、ここでは、法多項式ｆ（ｘ）の逆数β（ｘ）と被除多項式ｃ’（ｘ）とを乗算して前述した商を求める機能をもっている。

【０１２５】

具体的には、商立て回路５０は、２の拡大体演算制御部２２ａにより制御され、図１１に示すように、剰余算の１回目のみメモリ２内の法多項式ｆ（ｘ）の上位２ブロック（FL-1(x)，FL-2(x)）を逆数計算部５１に与えてその上位２ブロックの逆数β（ｘ）を算出させる機能と、得られたβ（ｘ）がメモリ２に書込まれると、メモリ２からこの逆数を読出す機能と、読出した逆数β（ｘ）と現在の被除多項式の上位２ブロック（C’L-1(x)，C’L-2(x)）とを乗算して商γ(x)を得る機能と、得た商γ(x) の上位２ブロック目を商ｑｉ(x)として立て、この商ｑｉ(x)をメモリ２に書込む機能と、これら逆数β（ｘ）の読出から商ｑｉ(x)の書込までの動作を剰余ｃ（ｘ）を得るまで繰り返す機能とをもっている。

【０１２６】

逆数計算部５１は、図１２に示すように、メモリ２内の法多項式ｆ（ｘ）の上位２ブロック（FL-1(x)，FL-2(x)）を商立て回路５０から受けると、図１３に示すように、この２ブロック（FL-1(x)，FL-2(x)）に基づいてその逆数β（ｘ）を算出する機能と、得られた逆数β（ｘ）をメモリ２に書込む機能とをもっている。また、逆数計算部５１は、その一部に図２７に示したＬＦＳＲが割り算回路として使用されている。

【０１２７】

ここで、逆数β（ｘ）は、ビット数が固定長であり、後段の除算処理本体にて除数の正規化並びに演算結果の逆正規化の処理を不要とする観点から、単なる逆数ではなく、図１３に示したように予め補正されている。また、逆数β（ｘ）自体は、逆数計算部５１を含む商立て回路５０に代えて、演算部４に算出させてもよい。

【０１２８】

なお、逆数計算部５１は、例えば整数型の積和演算回路のバス幅が８ビットのように小さいとき、全ての８ビット値の逆数値をテーブルにしてＲＯＭなどに記憶させる方式に置き換えが可能である。しかしながら、逆数計算部５１は、バス幅が１６ビット以上のとき、コストを低減させる観点から、全ての１６ビット値の逆数値をＲＯＭに記憶させる方式よりも好ましい。

【０１２９】

次に、以上のように構成された演算装置（コプロセッサ）の動作を説明する。

【０１３０】

本発明に係る２の拡大体多項式基底の剰余乗算は、乗算と剰余算とを別に行う。すなわち、図１４に示すように、乗算対象の多項式ａ(x)，ｂ(x) 及び法多項式ｆ(x) を入力し（ＳＴ１）、ａ(x) ・ｂ(x) の乗算を行って２倍のビット長の乗算結果C’（ｘ）を得た後（ＳＴ２）、C’（ｘ）ｍｏｄ　ｆ(x) の剰余算を行い（ＳＴ３）、剰余ｃ（ｘ）を得る（ＳＴ４）。

【０１３１】

ここで、ステップＳＴ２の乗算は、第１及び第２の実施形態で述べた通りであ

る。よって、ここではステップＳＴ３～ＳＴ４の剰余算の動作について説明する。なお、始めに筆算について延べ、次いで、筆算に対応した実際の処理を説明する。

【０１３２】

（６）式の剰余算は、図１５に筆算を示すように、除数ｆ(x) と、被除数C(x)とが所定ビット数ｋの単位ブロック毎に分割されて行われる。なお、単位ブロックのビット数は、例えばコプロセッサ１のバス幅に対応したビット数が適用可能である。

【０１３３】

次に、被除数ｃ(x) の上位ブロックｃL-i(x)がｆ(x) で除算され、上位桁から１ブロックの商qi(x) が立てられ、ｃ(x) －f(x)・ｑi(x)として、上位桁から１ブロックの被除数c(x)が減らされる。

【０１３４】

詳しくは、１ブロックの商qi(x) と除多項式f(x)との乗算結果が毎回（ｍ＋１）ブロックになる。また、この乗算結果を今回の被除多項式ｃ(x) から減算（＝加算）して、（２ｍ－１＊ｎ）ブロックの次回の被除多項式を算出し（ｎは乗算回数）、すなわち前回の被除数c(x)を１ブロックずつ減らしていく。

【０１３５】

剰余算は、このような商立てから減算までの処理をｎ回（＝被除数のビット数／単位ブロックのビット数）繰り返し、剰余ｃ（ｘ）を得て完了する。

【０１３６】

続いて、剰余算を行う実際の処理動作を説明する。

上述した剰余算において、商ｑi(x)を立てる動作は、図１１に示すように商立て回路５０が行い、ｃ(x) －f(x)・ｑi(x)として被除数ｃ(x) を減らす動作は、図１６に示すように演算部４が行う。これら商立て回路５０及び演算部４の動作について、以下、順次述べる。

【０１３７】

商立て回路５０は、最初の１回目の商計算を行うとき、図１１及び図１２に示すように、除数ｆ（ｘ）の逆数β（ｘ）を算出するため、メモリ２から除数ｆ(x

）の上位２ブロック（$F_{L-1}(x)$，$F_{L-2}(x)$）を読出すと、逆数計算部５１に入力する。

【０１３８】

逆数計算部５１は、図１３及び（８）式に示すように、この２ブロック（$F_{L-1}(x)$，$F_{L-2}(x)$）のうち、上位１ブロック$F_{L-1}(x)$の最上位ビットＭＳＢから連続した０の数をｄとして記憶する。

【０１３９】

$d = count\_zero(F_{L-1}(x))$　　…（８）

但し、count_zero()：()の値でＭＳＢから連続した０の数を数える関数

また、この無効な桁数ｄに基づいて、後述する左シフトの桁数ｈを（９）式のように算出して記憶する。

【０１４０】

$h = (d+1) \bmod k$　　…（９）

次に、逆数計算部５１は、図１３及び（１０）式に示すように、除数ｆ（ｘ）の上位２ブロック（$F_{L-1}(x)$，$F_{L-2}(x)$）の逆数α（ｘ）をＬＦＳＲ９０にて算出する。

【０１４１】

$\alpha(x) = x^{2k} / (F_{L-1}(x) \cdot X^{k} + F_{L-2}(x))$　　…（１０）

例えば、１ブロックが１６ビット（ｋ＝１６）の場合を説明する。また、被除数は、最上位ビットＭＳＢが“１”で他のビットが“０”のx^{2*16}（$=x^{2k}$）であるとする。

【０１４２】

逆数計算部５１は、上位２ブロック（$F_{L-1}(x)$，$F_{L-2}(x)$）を除数として図２７中の係数器９３に設定した後、被除数x^{2k}を高次からシフトレジスタに入力して１クロック毎のシフトを２＊１６回繰り返し、３２ビットの逆数α(x) を得る。なお、１ブロックは、８ビットや３２ビットでもよく、その他の任意のビット数でも同じ方式で逆数α（ｘ）を算出可能となっている。

【０１４３】

続いて、逆数計算部５１は、この逆数α（ｘ）をそのＭＳＢ側に（ｋ－１）ビ

ットの“０”と１ビットの“１”を連結して２ｋビットの値α’（ｘ）にする。その後、この２ｋビットの値α’（ｘ）を、（１１）式に示すように、（９）式で得た左シフトの桁数ｈだけ左にビットシフトし、補正された逆数β（ｘ）を算出する。

【０１４４】

$$\beta(x)=\alpha'(x)\cdot x^{h} \quad \cdots(11)$$

ここで、補正された逆数β(x) とは、以上の（８）式～（１１）式を満たす値である。また、逆数β（ｘ）は、与えられた法多項式ｆ(x) に対して１回だけ算出してメモリ２に保持し、以後はメモリ２から読み出される。また、被除数が変わっても、法多項式ｆ(x) が同一であれば逆数β（ｘ）も同一であるため、逆数β（ｘ）を算出せずにメモリ２から読出せばよい。

【０１４５】

商計算のうち、逆数β(x) が事前設定されていると、剰余算は、以下の（１２）式～（１５）式で実行できる。

【０１４６】

すなわち、商立て回路５０は、（１２）式及び図１１に示すように、現在の被除数Ｃｉ（０≦ｉ≦ｎ）の上位２ブロック（CL-1(x)，CL-2(x)）と逆数β(x) を掛ける。

$$\gamma(x)=\beta(x)\cdot(CL\text{-}1(x)\cdot x^{k}+CL\text{-}2(x)) \quad \cdots(12)$$

また、商立て回路５０は、（１３）式に示すように、この結果γ(x) のうち、１ブロック分の商ｑｉ(x)に該当する桁を上位２ブロック目として切出し処理し、メモリ２に書込む。

$$qi(x)=\gamma(x)/x^{2k} \quad \cdots(13)$$

これにより、１ブロック分の商ｑｉ(x)が得られる。

次に、現在の被除数ｃｉ(x)から除数と商の積ｆ(x) ・ｑｉ(x)を減算する処理は、図１６に示すように、演算部４が実行する。

【０１４７】

すなわち、演算部４では、２の拡大体上乗算回路１２が、法多項式ｆ(x) と、現在の１ブロック分の商ｑｉ(x)とを乗算して（１４）式に示すように乗算結果Ｐ

(x) を得る。

【０１４８】

$P(x) = f(x) \cdot qi(x)$ …（１４）

また（１５）式に示すように、加算回路１４，１５が、この乗算結果Ｐ(x) を現在の被除数Ｃｉ から減算し（= 加算し）、次回の被除数Ｃｉ+1 を得る。

$Ci+1 = Ci + P(x)$ …（１５）

以下、（１２）式～（１５）式をｎ回、繰り返して行い、最終的に、図１４～図１６に示す如き、剰余ｃ(x) を得る。この剰余ｃ(x) （＝ $[c_{m-1}, \cdots, c_1, c_0]$ ）が、（３）式に示した最終的な剰余乗算結果ｃ(x) に相当する。

【０１４９】

以上により、第１又は第２の実施形態に述べた（５）式の乗算結果ｃ(x) から（６）式の剰余算結果ｃ(x) を算出でき、もって、乗算及び剰余算からなる剰余乗算を完結することができる。

【０１５０】

（評価）

続いて、以上のように剰余乗算を行う第１及び第３の実施形態におけるコプロセッサ１の処理速度と回路規模を評価したので、順次説明する。

（処理速度の評価）

コプロセッサ１における各コマンドの所要クロック数は、ビット数ｍ＝１６０の場合とｍ＝１０２４の場合に関し、図１７に示す通りである。なお、楕円曲線暗号への応用では、ビット数ｍ＝１６０が典型的なサイズである。図示したｍ＝１０２４の場合は整数型ＲＳＡ方式暗号の現在安全であるとされる最大の鍵長が１０２４ビットであることから、将来想定される楕円曲線暗号への鍵長の増加に伴う速度の見積もりとして示した。

【０１５１】

次に、処理速度の相対比較のため、１６０ビットの２の拡大体ＧＦ（2^{160}）の加算、乗算、二乗算における処理クロック数を評価した。結果は図１８に示す通りである。なお、乗算と二乗算と乗算のクロック数は、ＧＦ（2^{160}）演算の速度比較のため、図１７とは異なり、法多項式による剰余算を含むクロック数と

なっている。

【０１５２】

また、相対比較値としてのＳＲ比は、コプロセッサ１のクロック数を通常のシフトレジスタ回路のクロック数で除した値であり、この値が小さいほど処理速度が高速であることを示す。このＳＲ比によれば、本発明のコプロセッサ１は、加算を除き、通常のシフトレジスタ回路と同等の処理速度で２の拡大体演算を実行できることが分かる。

【０１５３】

（回路規模の評価）

コプロセッサ１の回路規模は、図１９に示すように、全体で約３０ｋゲートの規模となる。このコプロセッサ１の回路は、整数型コプロセッサに対し、２の拡大体演算を処理するための回路を追加したものである。

【０１５４】

具体的には図２０に示すように、演算部４では、積和演算回路において、キャリー有り・無しの切替回路を追加している。制御部５では、加算、乗算及び二乗算に関し、追加がほぼ不要であるが、除算に関し、商立て回路５０を追加している。ＲＡＭ（メモリ２）及びＩ／Ｆは、整数型コプロセッサと共用するため追加が不要である。

【０１５５】

これにより、追加回路の規模は、全体で約５ｋゲートとなる。５ｋゲートという追加回路量は、最近のＬＳＩ技術では大きい量ではなく、現行コプロセッサに代えて、本発明のコプロセッサ１を用いることが十分可能な範囲にある。

【０１５６】

また比較のため、本発明のコプロセッサ１を用いずに２の拡大体の演算機能（加算・乗算・二乗算）を実現する場合における２の拡大体演算専用のコプロセッサの回路規模を見積もった。結果は、図２１に示す通りである。

【０１５７】

図示するように、２の拡大体演算専用のコプロセッサの回路規模は、ｍ＝１６０の場合に１０ｋゲートであり、ｍ＝１０２４の場合には１６ｋゲートである。

これにより、２の拡大体の演算機能を実現する場合、本発明に係るコプロセッサ１を用いた方が、２の拡大体演算専用のコプロセッサを設ける場合に比べ、約１／２～１／３という少ない追加回路で実現できることが分かる。

【０１５８】

上述したように本実施形態によれば、第１の実施形態の効果に加え、剰余算処理の際に線形フィードバックシフトレジスタＬＦＳＲ９０ではなく、多倍長の積和演算回路が演算を行うので、１以上の任意の拡大次数ｍを使用でき、２の拡大体の拡大次数ｍを増加しても、装置本体を作り直さずに演算を実行することができる。さらに、拡大次数ｍの制限によるハードウェア的な制約を無くすことで暗号鍵のビットの増加にも対応することができる。

【０１５９】

また、２の拡大体の剰余乗算を乗算処理と剰余算（除算）処理とに分割し、任意の法多項式ｆ(x) を使用できるようにしたので、汎用性を向上させることができる。

【０１６０】

また、商立て回路５０が、剰余算処理のとき、被除多項式ｃ(x) と除多項式ｆ(x) とに基づいて商計算を行い、上位からバス幅と同じビット数の１ブロックの商多項式ｑｉ(x)を立てると、演算部４が、この商多項式ｑｉ(x)と除多項式ｆ(x) との乗算結果ｑｉ(x)・ｆ(x) を今回の被除多項式ｃｉ(x)から減らして次回の被除多項式ｃｉ-1(x)を算出する。

【０１６１】

コプロセッサ１は、このような商立て回路５０による商計算と、演算部４の積和演算による被除多項式データの算出までの処理を繰り返して剰余ｃ（ｘ）を得るので、ハードウェアの特性を生かして剰余算と商計算の効率化を図ることができる。

【０１６２】

さらに、商立て回路５０としては、商計算のとき、除多項式データの上位２ブロックの逆数データと今回の被除多項式データの上位２ブロックとを乗算し、この乗算結果の上位２ブロック目を１ブロックの商多項式データとすることにより

、得られた商多項式のうち、有効数字の部分を抽出できるので、演算精度の最適化を図ることができる。

【０１６３】

また、商計算において、除多項式ｆ(x)の上位２ブロックからの逆数β(x)の算出をコマンドとして独立させ、２の拡大体演算に先行して逆数β(x)を算出し、得られたβ(x)をメモリ２に格納し、剰余算の実行時には逆数β(x)をメモリ２から読出す。

【０１６４】

すなわち、同じ法多項式下での重複する剰余算の実行時に、逆数データをメモリから読出して商を立てるので、２回目の商計算以降は逆数データの算出時間を省略でき、もって、２の拡大体乗算（剰余乗算）、二乗算の処理時間を短縮することができる。また、逆数β(x)を事前に算出できるので、２の拡大体の剰余乗算を、乗算と加算とを行う積和演算回路のみを用いて実現することができる。

【０１６５】

次に、商立て回路５０としては、逆数データを算出するとき、除多項式データの上位２ブロックのうち、上位から連続した０の数を計数すると、上位から１ブロック＋１ビットの多項式データを最上位ビットを１とするように抽出し、この抽出した多項式データの逆数を求め、得られた逆数の最上位ビット側に、最下位ビットが１で他のビットが０の１ブロックの補正データを連結して全体で２ブロックのデータを求め、このデータを前記計数した０の数だけ上位側にビットシフトさせた結果を逆数データとする。

【０１６６】

このため、一般的な多倍長の整数型の除算で用いられる単精度除算を用いたKnuth のアルゴリズムによる除数の正規化や近似商の補正処理、商や剰余といった演算結果の逆正規化の処理を行わないように、予め補正した値を逆数データとしたので、ビットシフトの回数を減らすことができ、演算装置を最適化することができる。

【０１６７】

例えば、整数型の乗算では、ｍビット＊ｍビット＝２ｍビットとなり、２ｍビ

ットの上位に数ビットの連続した０が並んでいる場合でも有効なビット数は２ｍである。この乗算結果を用いて除算（剰余算）を行う場合、０での割り算はできないため、予め除数、被除数を左シフトしてＭＳＢに１が立つように正規化する必要がある。所定のループを終えて演算が終了した時点で、事前に左シフトしたビット数だけ演算結果（商、剰余）を右シフトする逆正規化の処理も必要である。

一方、本実施形態では、このような除算ループの前後処理を不要にする観点から、商計算時の除数（逆数データβ(x)）を補正したので、演算装置を最適化することができる。

【０１６８】

また、本実施形態は、ビット単位ではなく、ブロック単位で演算を実行し、且つ補正された逆数β(x)を用いて演算を実行するので、ビットシフトの回数を低減でき、処理速度の高速化を図ることができる。

【０１６９】

さらに、少ないコマンドと、多倍長積和演算回路を用いた演算方式により、通常のシフトレジスタ型の２の拡大体乗算回路と同等の処理速度を有し、且つ、整数型演算や２の拡大体演算に基づく各種暗号方式を実行可能なＬＳＩを登載した演算装置及び暗復号装置を少量の追加回路により実現することができる。なお、２の拡大体演算を用いた暗号方式としては、素体版楕円番号、多項式基底楕円暗号等の楕円曲線暗号が適用可能となっている。

【０１７０】

また、本実施形態は、第１の実施形態における除算過程の具体例として説明したが、第2の実施形態における除算過程の具体例としても、同様の作用効果を得ることができる。

【０１７１】

（第４の実施形態）

図２２は本発明の第４の実施形態に係る演算装置及び暗号処理装置に適用されるコプロセッサの構成例を示す模式図である。

【０１７２】

本実施形態は、第１～第３の実施形態の変形形態であり、２の拡大体演算専用の演算装置としたものであって、具体的には、整数型乗算器１１、セレクタ１３、整数演算制御部２１を省略した構成となっている。但し、演算アルゴリズムは、前述した通りであり、２の拡大体の乗算処理を乗算と剰余算とに分割し、乗算の後に剰余算を実行する。

【０１７３】

以上のような構成としても、整数型演算自体の作用効果と、整数型演算並びに２の拡大体演算の切替に関する作用効果とを除き、第１～第３の実施形態の効果を得ることができる。換言すると、２の拡大体演算に関し、第１～第３の実施形態の効果を得ることができる。

【０１７４】

その他、本発明はその要旨を逸脱しない範囲で種々変形して実施できる。

【０１７５】

【発明の効果】

以上詳記したように本発明によれば、最小のアーキテクチャを追加するだけで整数型の演算に加えて２の拡大体上の演算をも実行できる演算装置及び暗号処理装置を提供することができる。

【０１７６】

また、２の拡大体の拡大次数ｍを増加しても、装置本体を作り直さずに演算を実行できる演算装置及び暗号処理装置を提供できる。

【図面の簡単な説明】

【図１】

本発明の第１の実施形態に係る演算装置の構成例を示すブロック図。

【図２】

ｃ’（ｘ）＝ａ（ｘ）＊ｂ（ｘ）を実現するための４＊４ビットの単位乗算の回路構成例を示す図。

【図３】

整数型乗算を実現するための４＊４ビットの単位乗算の回路構成例を示す図。

【図４】

同実施形態におけるコプロセッサに用いられるキャリー制御機能付きの４ビットのリップルキャリー型全加算器の構成例を示すブロック図。

【図５】

同実施形態の加算回路に用いられる全加算器及びキャリー制御スイッチの構成例を示す図。

【図６】

キャリー制御機能付き全加算器の変形例を示す図。

【図７】

キャリー制御機能付き全加算器の他の変形例を示す図。

【図８】

本発明の第２の実施形態に係る演算装置の構成例を示すブロック図。

【図９】

同実施形態の乗算回路を実現するための４＊４ビットの単位乗算の回路構成例を示す図。

【図１０】

本発明の第３の実施形態に係る演算装置及び暗号処理装置に適用されるコプロセッサの構成例を示すブロック図

【図１１】

同実施形態における商立て回路の構成を示す模式図

【図１２】

同実施形態における逆数計算部の機能を説明するための模式図

【図１３】

同実施形態における逆数計算部の構成を示す模式図

【図１４】

同実施形態における２の拡大体多項式基底の剰余乗算方式を説明するためのフローチャート

【図１５】

同実施形態における剰余算処理を説明するための筆算の模式図

【図１６】

同実施形態における演算部の処理を示す模式図

【図１７】

同実施形態における各コマンドの所要クロック数を示す図

【図１８】

同実施形態におけるＧＦ(2^{160})演算の所要クロック数を示す図

【図１９】

同実施形態におけるコプロセッサの回路規模を示す図

【図２０】

同実施形態における追加回路量を示す図

【図２１】

同実施形態における比較用のＧＦ(2^{m}) 演算専用のコプロセッサの回路規模を示す図

【図２２】

本発明の第４の実施形態に係る演算装置及び暗号処理装置に適用されるコプロセッサの構成例を示す模式図

【図２３】

暗号処理用演算用コプロセッサを含むＩＣカード用ＬＳＩのブロック図。

【図２４】

図１０に示すＬＳＩにおけるコプロセッサ部分の構成例を示すブロック図。

【図２５】

多項式基底の２の拡大体演算を行うコプロセッサのハードウェア構成例を示すブロック図。

【図２６】

円分体といわれる２の拡大体の乗算回路を示す図。

【図２７】

一般的な線形フィードバックシフトレジスタＬＦＳＲの構成を示すブロック図

【符号の説明】

１，１’…コプロセッサ

２…メモリ

３…データバス

１１…整数型乗算回路

１２…２の拡大体上乗算回路

１３…セレクタ

１４…加算回路

１５…加算回路

１６…キャリー保持部

２１…整数演算制御部

２２…２の拡大体演算制御部

３２…全加算器

３３…スイッチ

４１…乗算回路

４２，４３，４４…キャリー制御機能付き全加算器

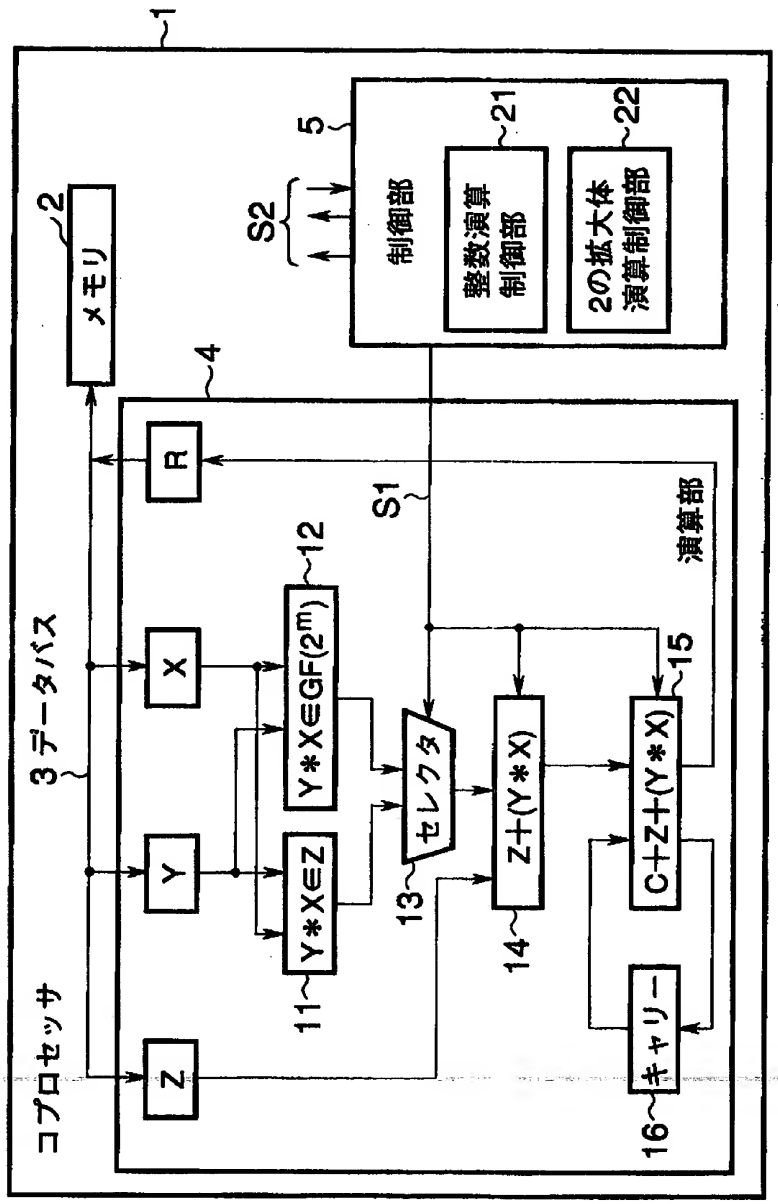
５０…商立て回路

５１…逆数計算部

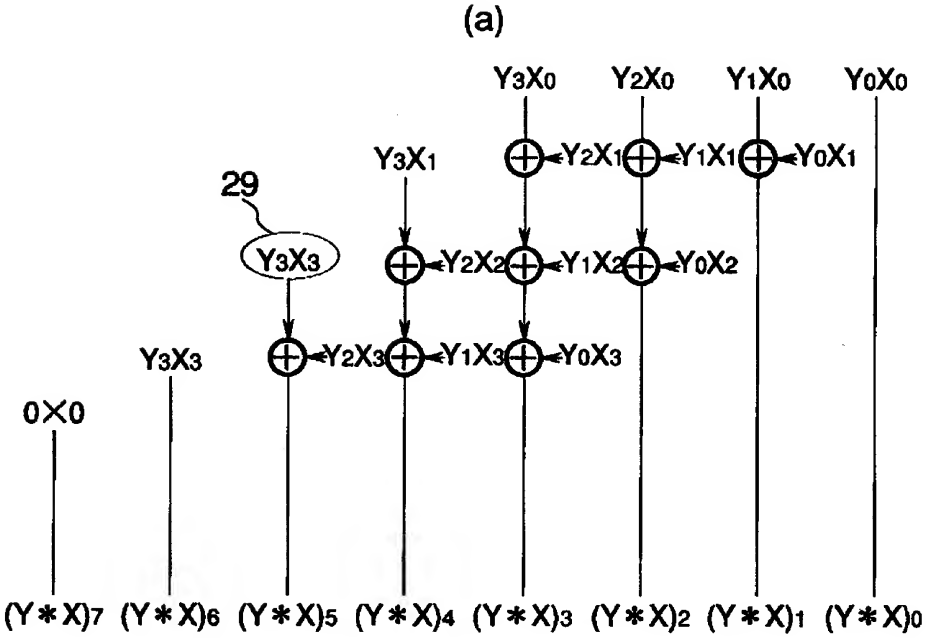
Ｓ１，ｓ２…制御信号

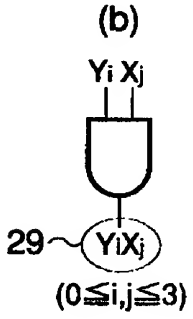
【書類名】　図面

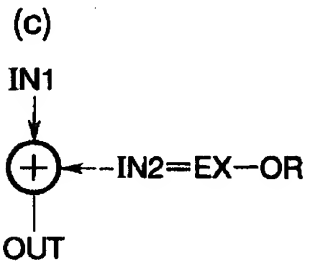
【図１】


1
2
メモリ
3 データバス
4
5
制御部
21
整数演算制御部
22
2の拡大体演算制御部
S2
S1
演算部
R
X
Y
Z
12
Y*X∈GF(2^m)
11
Y*X∈Z
13
セレクタ
14
Z+(Y*X)
15
C+Z+(Y*X)
16
キャリー
コプロセッサ

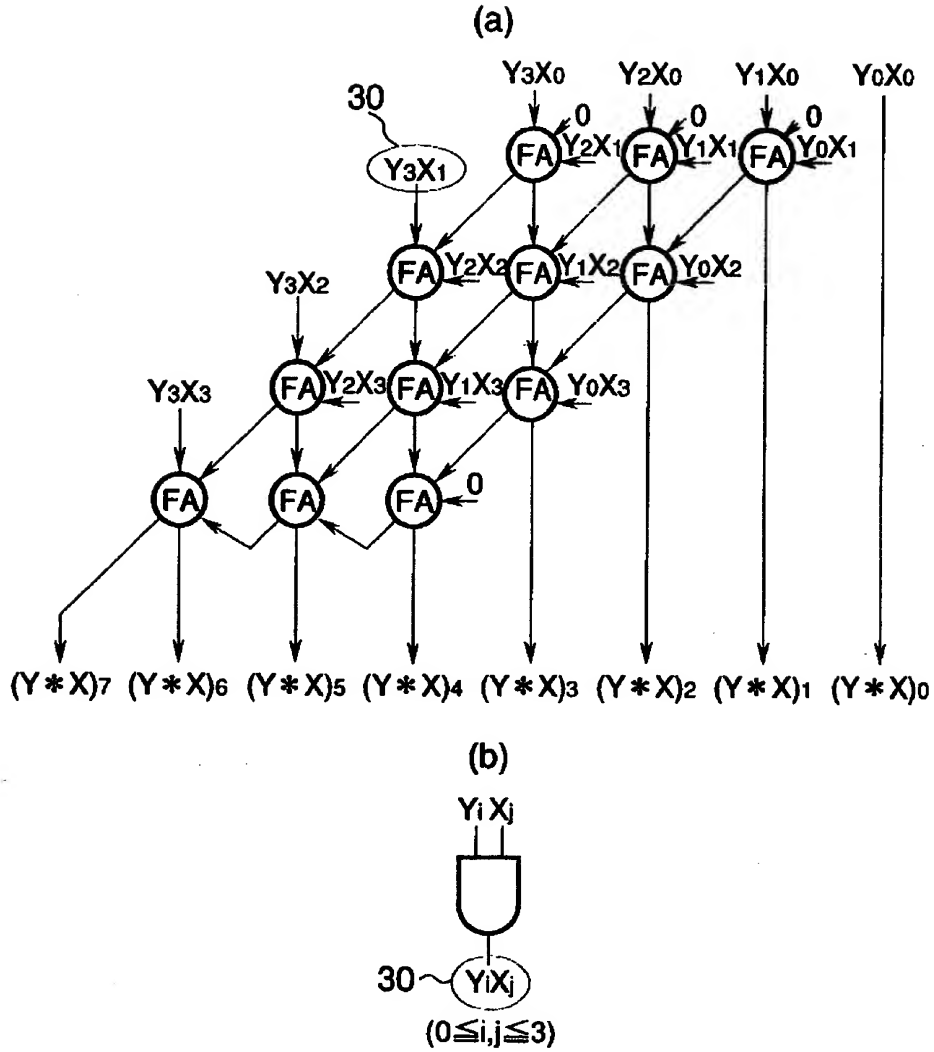
【図２】


(a)
Y3X0
Y2X0
Y1X0
Y0X0
Y3X1
Y2X1
Y1X1
Y0X1
29
Y3X3
Y2X2
Y1X2
Y0X2
Y3X3
Y2X3
Y1X3
Y0X3
0×0
(Y＊X)7
(Y＊X)6
(Y＊X)5
(Y＊X)4
(Y＊X)3
(Y＊X)2
(Y＊X)1
(Y＊X)0

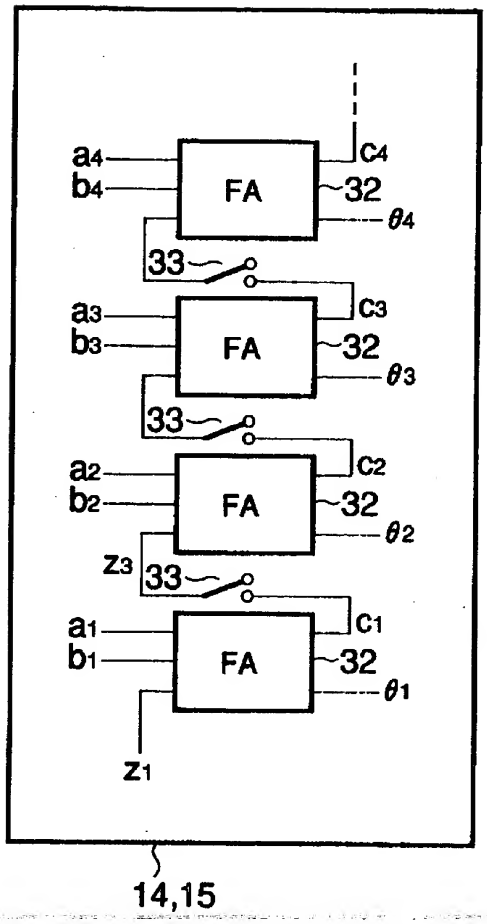

(b)
Yi Xj
29
YiXj
(0≦i,j≦3)


(c)
IN1
IN2=EX−OR
OUT

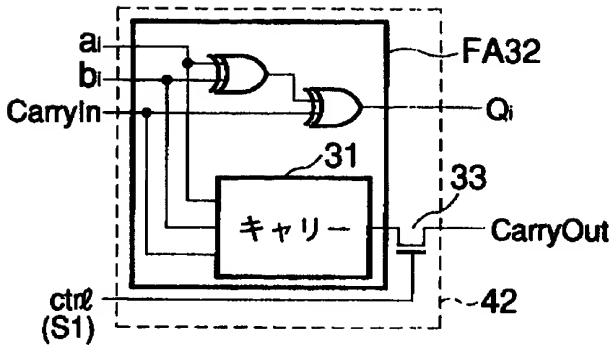
【図３】


(a)
Y3X0
Y2X0
Y1X0
Y0X0
30
Y3X1
FA
Y2X1
Y1X1
Y0X1
Y2X2
Y1X2
Y0X2
Y3X2
Y2X3
Y1X3
Y0X3
Y3X3
0
(Y＊X)7
(Y＊X)6
(Y＊X)5
(Y＊X)4
(Y＊X)3
(Y＊X)2
(Y＊X)1
(Y＊X)0
(b)
Yi Xj
30
YiXj
(0≦i,j≦3)
(c)
FA
ai
bi
CarryIn
Qi
キャリー
Carry
Out
31
(d)
ai
bi
CarryIn
Carry
Out
31 キャリー

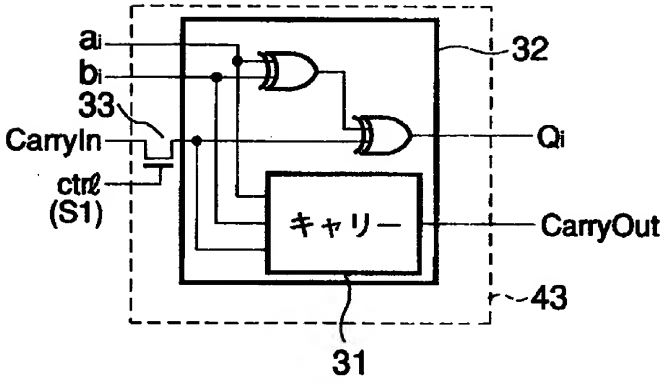
【図４】


a4
b4
FA
C4
32
θ4
33
a3
b3
FA
C3
32
θ3
33
a2
b2
FA
C2
32
θ2
z3
33
a1
b1
FA
C1
32
θ1
z1
14,15

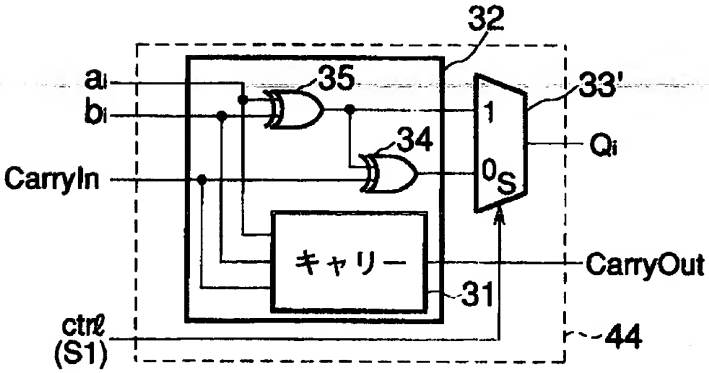
【図５】


ai
bi
CarryIn
FA32
Qi
31
33
キャリー
CarryOut
ctrℓ
(S1)
42

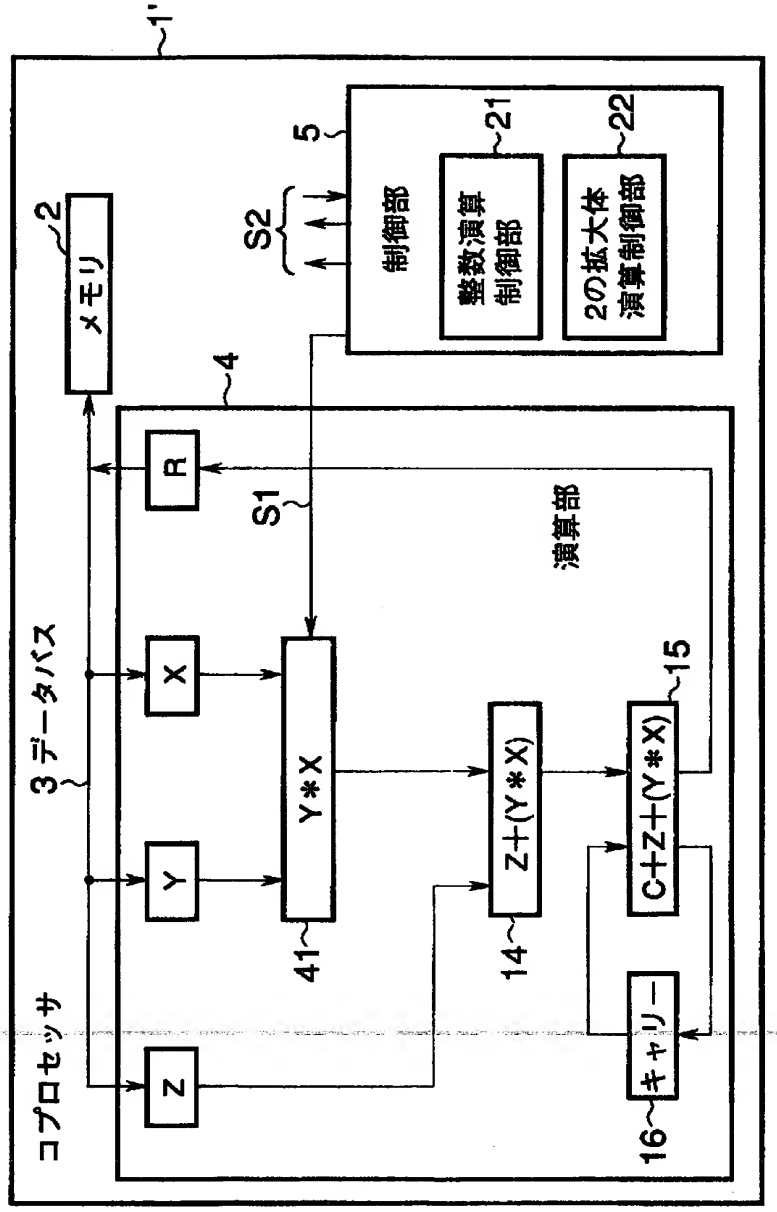
【図６】


ai
bi
33
CarryIn
ctrℓ
(S1)
32
Qi
キャリー
CarryOut
43
31

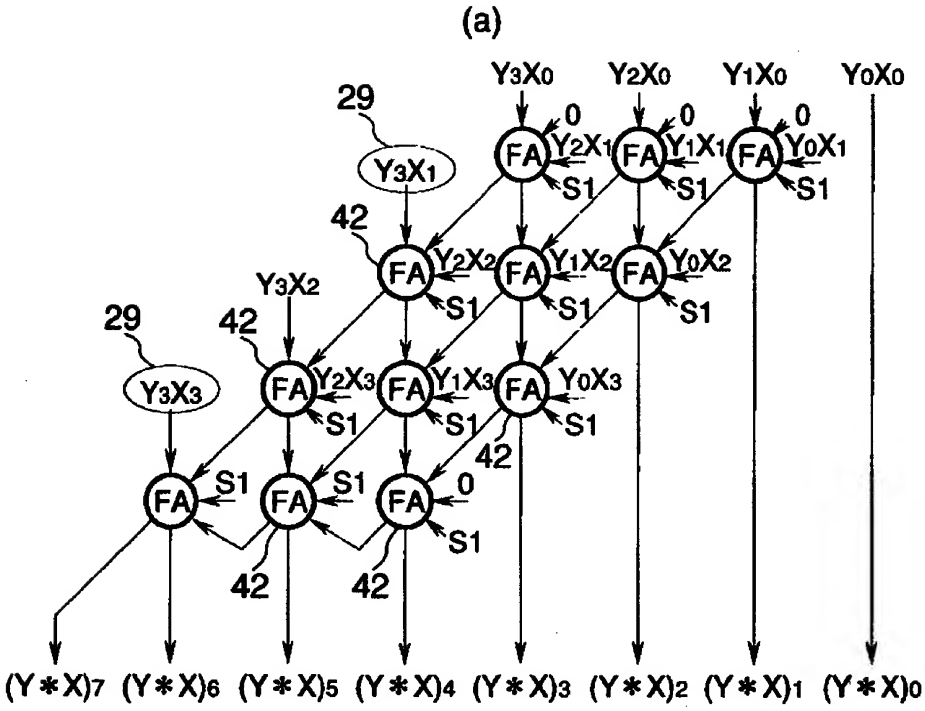
【図７】

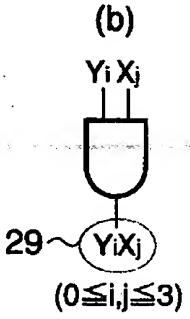

32
ai
bi
35
33'
1
Qi
34
CarryIn
0
S
キャリー
CarryOut
31
ctrℓ
(S1)
44

【図８】

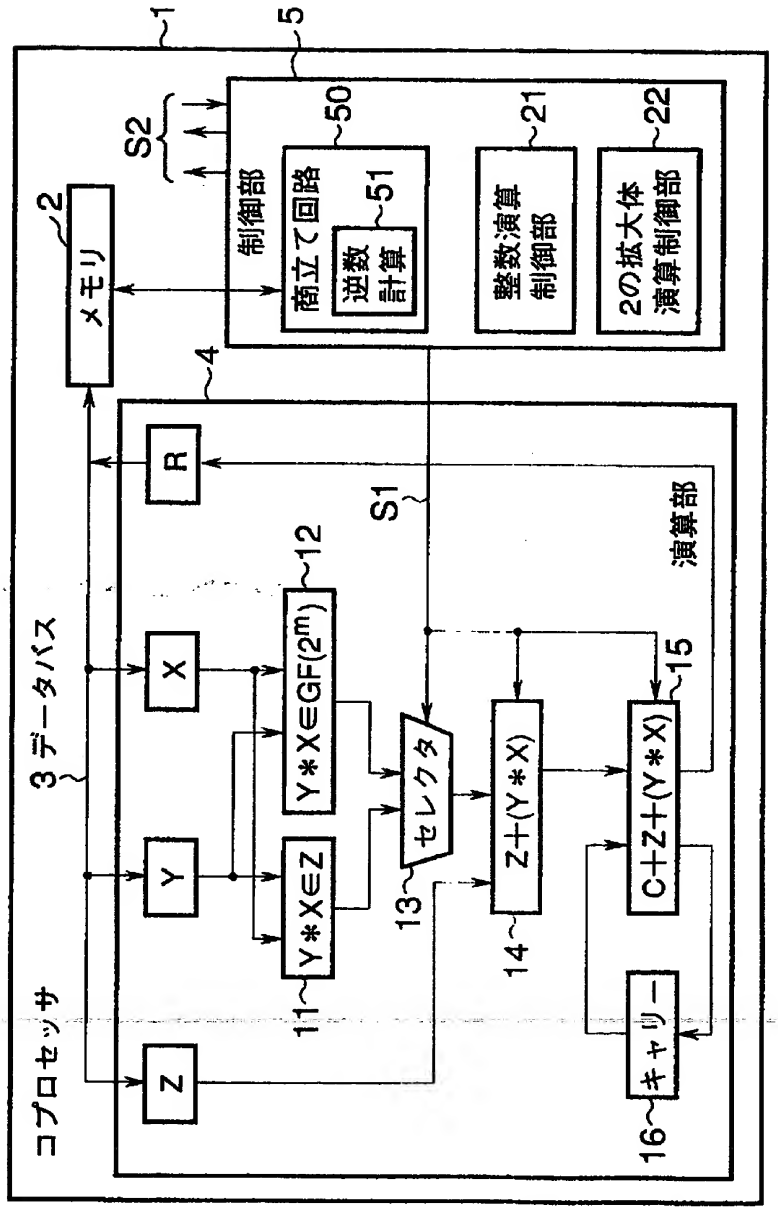

1'
2
メモリ
3 データバス
コプロセッサ
4
演算部
R
X
Y
Z
S1
Y*X
41
Z+(Y*X)
14
C+Z+(Y*X)
15
キャリー
16
S2
5
制御部
21
整数演算制御部
22
2の拡大体演算制御部

【図９】

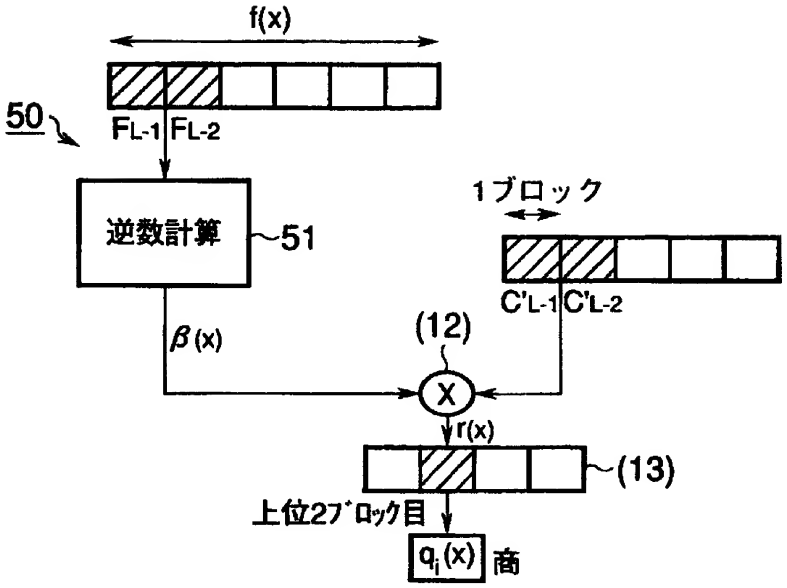

(a)
Y3X0
Y2X0
Y1X0
Y0X0
29
Y3X1
42
Y3X2
29
Y3X3
FA
S1
0
Y2X1
Y1X1
Y0X1
Y2X2
Y1X2
Y0X2
Y2X3
Y1X3
Y0X3
42
(Y*X)7
(Y*X)6
(Y*X)5
(Y*X)4
(Y*X)3
(Y*X)2
(Y*X)1
(Y*X)0


(b)
Yi Xj
29
YiXj
(0≦i,j≦3)

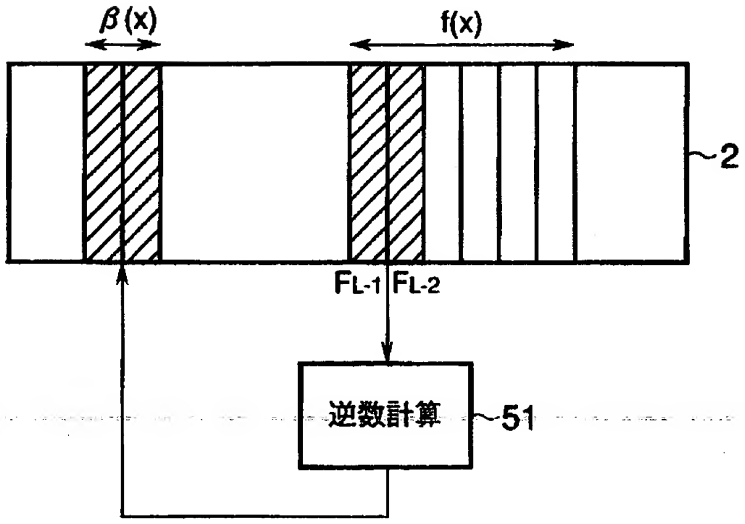
【図１０】


1
5
S2
50
制御部
商立て回路
51
逆数計算
21
整数演算制御部
22
2の拡大体演算制御部
2
メモリ
4
R
S1
演算部
12
X
Y*X∈GF(2^m)
3 データバス
Y
Y*X∈Z
11
13
セレクタ
14
Z+(Y*X)
15
C+Z+(Y*X)
Z
16
キャリー
コプロセッサ

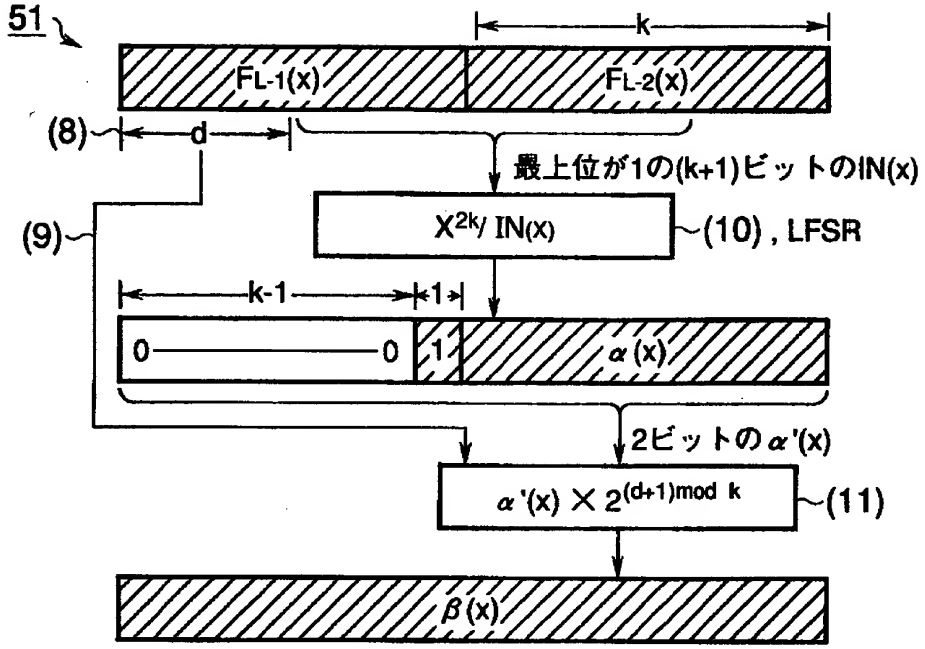
【図１１】


f(x)
50
FL-1 FL-2
逆数計算
51
1ブロック
C'L-1 C'L-2
β(x)
(12)
X
r(x)
(13)
上位2ﾌﾞﾛｯｸ目
qi(x)
商

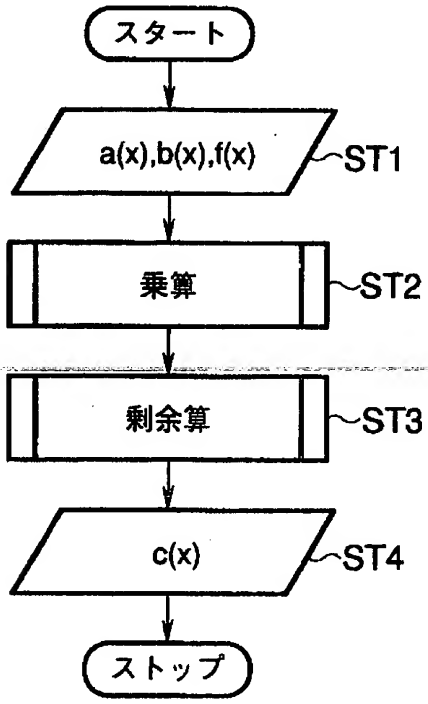
【図１２】


β(x)
f(x)
2
FL-1 FL-2
逆数計算
51

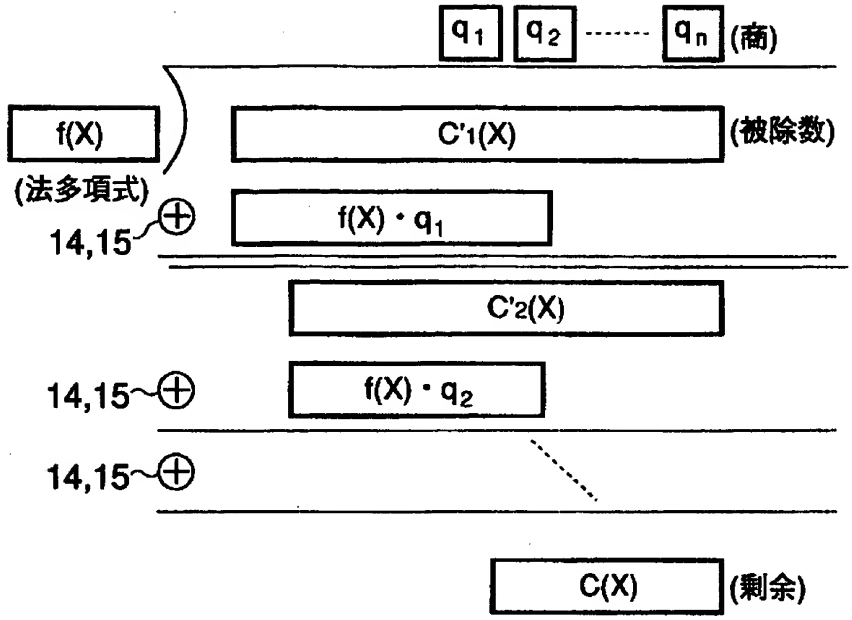
【図１３】


51
k
FL-1(x)
FL-2(x)
(8)
d
最上位が1の(k+1)ビットのIN(x)
(9)
X2k/ IN(x)
(10), LFSR
k-1
1
0
0
1
α(x)
2ビットのα'(x)
α'(x) × 2(d+1)mod k
(11)
β(x)

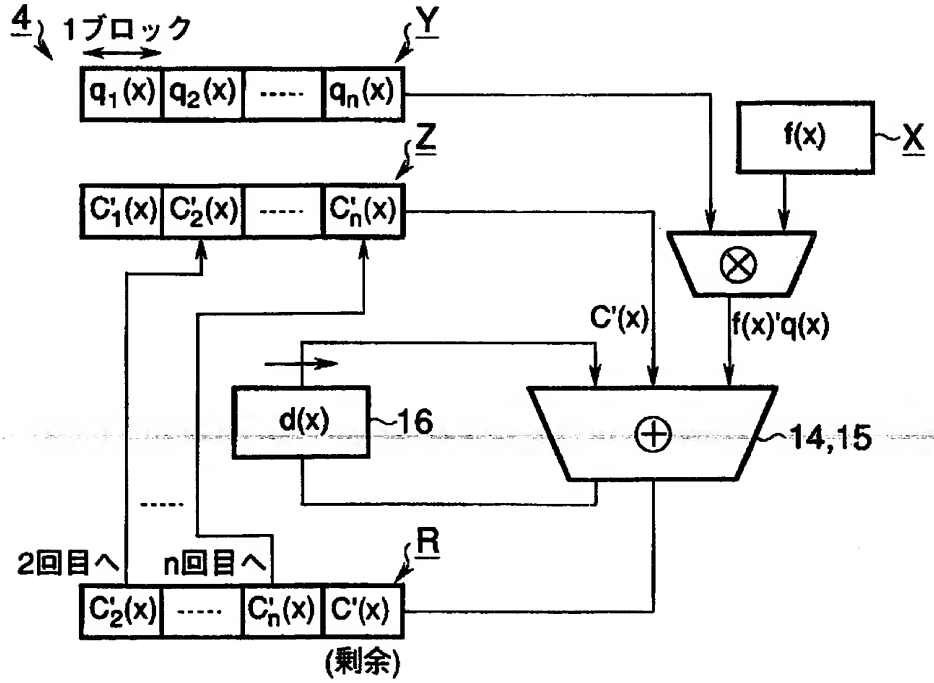
【図１４】


スタート
a(x),b(x),f(x)
ST1
乗算
ST2
剰余算
ST3
c(x)
ST4
ストップ

【図１５】


q1
q2
qn
(商)
f(X)
C'1(X)
(被除数)
(法多項式)
14,15
f(X)・q1
C'2(X)
14,15
f(X)・q2
14,15
C(X)
(剰余)

【図１６】


4
1ブロック
Y
q1(x)
q2(x)
qn(x)
Z
C'1(x)
C'2(x)
C'n(x)
f(x)
X
C'(x)
f(x)'q(x)
d(x)
16
14,15
2回目へ
n回目へ
R
C'2(x)
C'n(x)
C'(x)
(剰余)

【図１７】

各コマンドの所要クロック数

コマンド		m=160	m=1024
加算		14	68
乗算		64	2,116
二乗算		25	133
除算	事前計算	35	35
	本体	134	2,564

【図１８】

GF(2^{160})の所要クロック数

演算	クロック数	SR比
加算	14	約4.6倍
乗算	198	約1.2倍
二乗算	159	約1倍

(SR比)=(ｸﾛｯｸ数)/(ｼﾌﾄﾚｼﾞｽﾀ回路でのｸﾛｯｸ数)

【図１９】

本発明のコプロセッサ回路規模(ゲート数)

演算部	8k
制御部	12.8k
RAM	8.5k
I/F	0.5k
全体	約30k

【図２０】

整数型のコプロへの追加回路量(ゲート数)

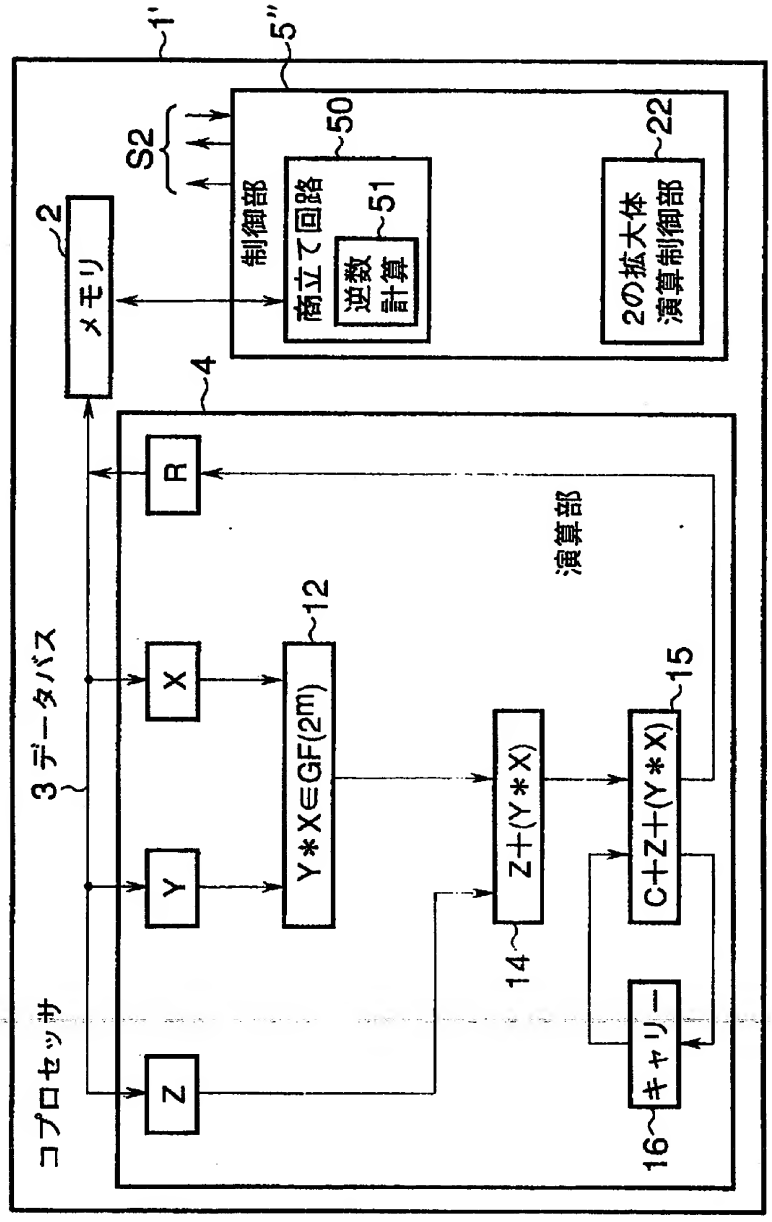
演算部	1k
制御部	3.8k
RAM	0(共用)
I/F	0(共用)
全体	4.8k

【図２１】

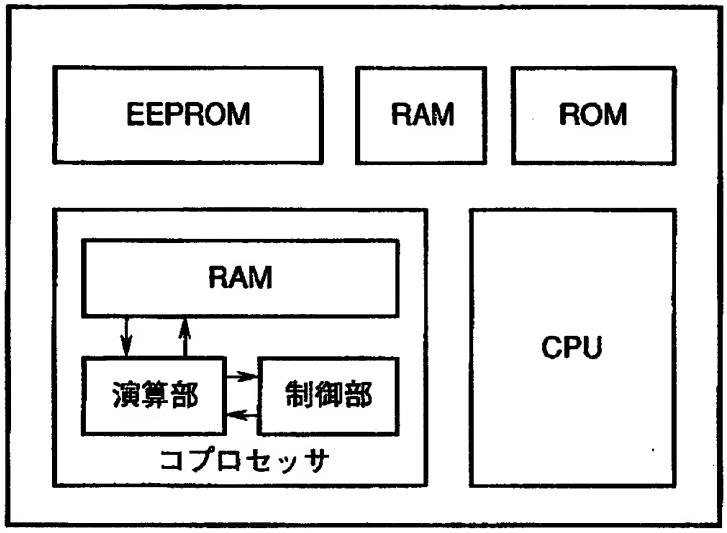
GF(2^m)単独回路規模(ゲート数)

	m=160	m=1024
演算部	3.1k	3.1k
制御部	3.8k	3.8k
RAM	2.3k	8.5k
I/F	0.5k	0.5k
全体	約10k	約16k

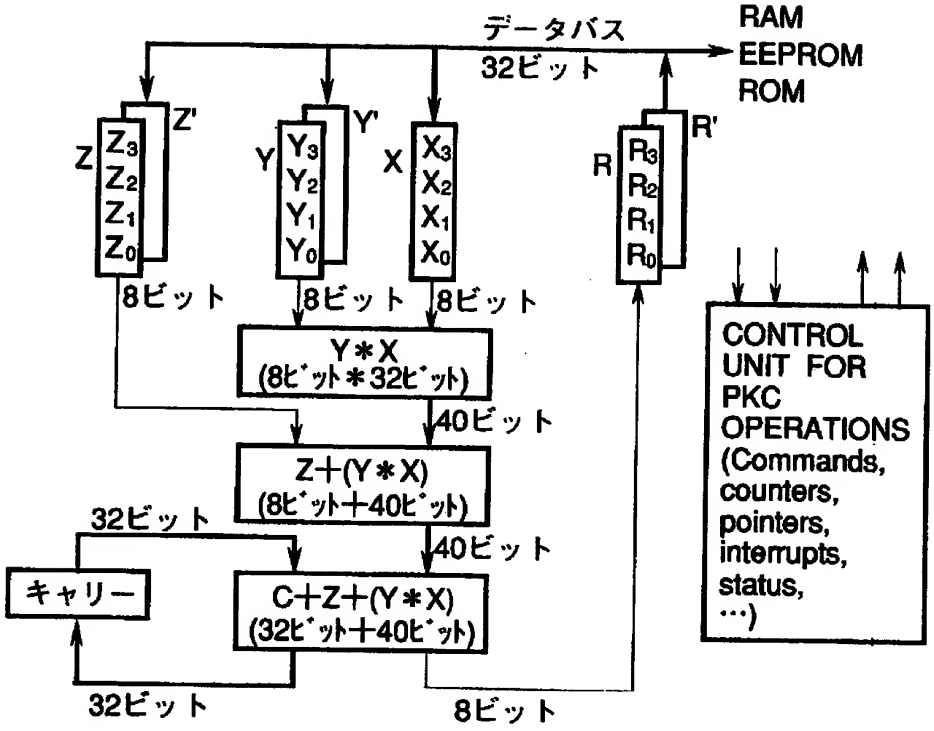
【図２２】


1'
5''
S2
50
制御部
商立て回路
逆数
計算
51
22
2の拡大体
演算制御部
2
メモリ
4
R
演算部
3データバス
X
12
Y*X∈GF(2^m)
15
Y
Z+(Y*X)
C+Z+(Y*X)
14
Z
キャリー
16
コプロセッサ

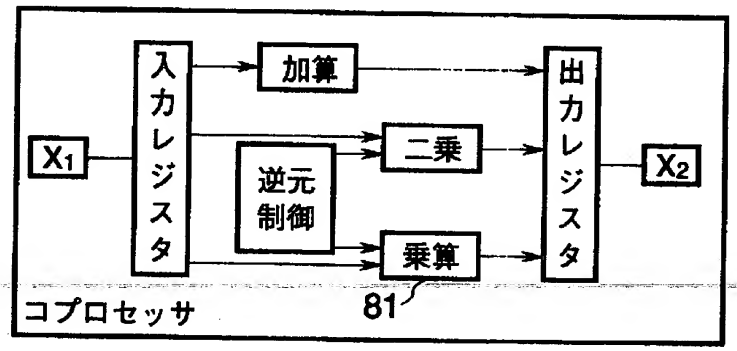
【図２３】



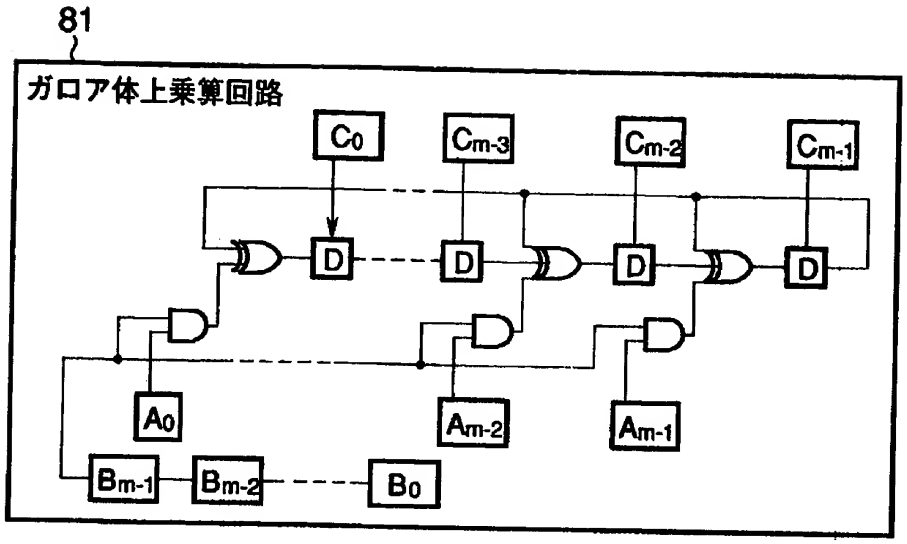
【図２４】


データバス
32ビット
RAM
EEPROM
ROM
Z
Z'
Y
Y'
X
R
R'
8ビット
8ビット
8ビット
Y＊X
(8ビット＊32ビット)
40ビット
Z+(Y＊X)
(8ビット+40ビット)
32ビット
40ビット
キャリー
C+Z+(Y＊X)
(32ビット+40ビット)
32ビット
8ビット
CONTROL
UNIT FOR
PKC
OPERATIONS
(Commands,
counters,
pointers,
interrupts,
status,
…)

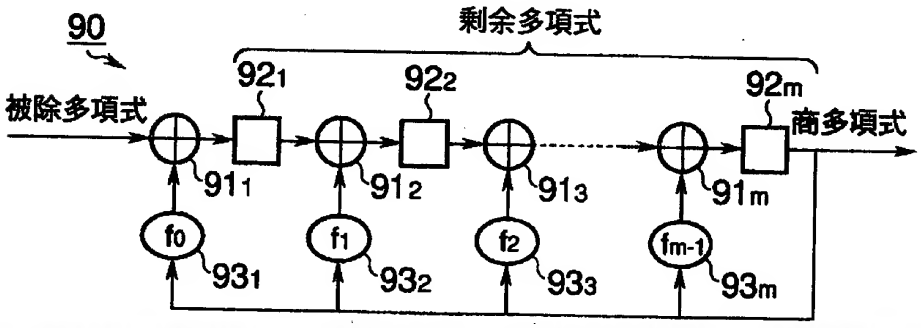
【図２５】


X1
入力レジスタ
加算
二乗
逆元制御
乗算
出力レジスタ
X2
コプロセッサ
81

【図２６】


81
ガロア体上乗算回路
C0
Cm-3
Cm-2
Cm-1
D
D
D
D
A0
Am-2
Am-1
Bm-1
Bm-2
B0

【図２７】


90
剰余多項式
被除多項式
921
922
92m
商多項式
911
912
913
91m
f0
f1
f2
fm-1
931
932
933
93m

【書類名】　　要約書

【要約】

【課題】　本発明は、２の拡大体の拡大次数ｍを増加しても、装置本体を作り直さずに演算の実行を図る。

【解決手段】　２の拡大体の多項式基底表現での剰余乗算を実行可能な多倍長の積和演算回路（１２，１４，１５）を有し、さらに、剰余乗算を乗算処理と剰余算処理とに分割して積和演算回路の制御により実行するための制御手段（５）を備えた演算装置及び暗号処理装置。

【選択図】　　図１

出願人履歴情報

識別番号　〔０００００３０７８〕

１．変更年月日　１９９０年　８月２２日
［変更理由］　新規登録
住　所　神奈川県川崎市幸区堀川町７２番地
氏　名　株式会社東芝